The Dissertation Committee for Savithri Sundareswaran
certifies that this is the approved version of the following dissertation:

Statistical Characterization For Timing Sign-Off: From Silicon to Design and Back to Silicon

Committee:

Jacob A. Abraham, Supervisor

David Blaauw

David Pan

Robert Flake

Michael Orshansky

Statistical Characterization For Timing Sign-Off: From Silicon to Design and Back to Silicon

by

Savithri Sundareswaran, B.E., M.S.

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

May 2009

Dedicated to my family.

Acknowledgments

My inexpressible gratitude goes to my advisor Prof. Jacob Abraham for not just the best advisor I could have desired, but also the visionary and enthusiastic researcher that has impacted me in several ways. I would also like to thank him for the research freedom and support that have made my PhD an enjoyable and smooth experience.

I would like to thank my mentor, Prof. David Blaauw, at the University of Michigan, who played an instrumental role in providing me with guidance and expertise during this research. My working experience with him at Motorola has been one of the most rewarding experiences and has enabled me to stay focused on being current with the technology.

I am thankful to my dissertation committee members, Prof. David Pan, Prof. Robert Flake and Prof. Michael Orshansky for taking the time out of their schedules to review my research and provide useful feedback.

All the research for this Ph.D. was done while working at Freescale Semiconductor. I would like to thank several co-authors and collaborators for their cooperation and providing timely feedbacks which was instrumental in making this research happen. Especially, I would like to thank Dr. Rajendran Panda for providing me with an enjoyable work environment at Freescale Semiconductor.

I am thankful to Alexandre Ardelea, Robert Maziasz, Yun Zhang, Lucie Nechanicka, Andy Hoover, Surya Veeraraghavan, Srinivas Jallepalli, Lionel Riviere for their insightful discussions and cooperation. I would like to thank Sergey Gavrilov and Roman Solovyev

for being great collaborators.

I would like to thank my husband, Sivaraman Saravanabavan for his patience, support and motivation without which I could not have finished this degree. I want to thank my sister Gayathri Sundareswaran for being my best friend and my family for providing me the inspiration.

Last, but not least, I would like to express my hearty gratitude to my mother for her unconditional love and support. For her sacrifices and constant encouragement which facilitated my earning this degree.

Statistical Characterization For Timing Sign-Off: From Silicon to Design and Back to Silicon

Publication No. _____

Savithri Sundareswaran, Ph.D.
The University of Texas at Austin, 2009

Supervisor: Jacob A. Abraham

With aggressive technology scaling, within-die random variations are becoming the most dominant source of process variations. Gate-level statistical static timing is becoming a widely accepted approach as an alternative to static timing analysis. However, statistical timing approaches lack good models for handling timing variations due to within-die random variations. Before performing statistical timing analysis on a design or System On Chip (SoC), the cells in the library are pre-characterized for delay as well as constraints due to these random variations. This is referred to as statistical characterization of the cells. The major contribution of this dissertation is the development of novel techniques for statistical characterization and optimization of cells. The methods couple the knowledge of circuits along with the significant factor analysis methods to compute the sensitivities, to perform statistical timing and to perform sensitivity-aware cell optimizations.

The first contribution of this dissertation is a statistical delay characterization method developed for computing delay sensitivities of standard cells considering both global and mismatch process variations. In addition to the cells being characterized for delay, the

sequential cells are characterized for timing constraints like setup and hold time constraints. The second contribution of this dissertation addresses the problem of constraint sensitivity characterization in sequential cells.

Block-based statistical timing approaches lack accurate consideration of the impact of slew variations on both delay and arrival time variations. Specifically, the delay variations due to within-die random variables (mismatch variables) result in a slew-based correlation during timing propagation. Handling within-die random variations more accurately during statistical timing propagation is the topic of the third contribution of this dissertation. Clock networks are more prone to these within-die random variations and can result in significant clock-skew variations. In the fourth contribution, a timing margining methodology is presented that accurately accounts for the clock skew variations in a timing sign-off flow.

Typically, the standard cells are designed very early in the design cycle and long before the process reaches production maturity. Any subtle improvements to reduce variability in standard cells can improve parametric yield significantly. Statistical characterization of cells for timing provides a key baseline for understanding the circuit behavior due to different sources of variation. The sensitivity information can also help increase yield by reducing the variability during the circuit design itself. The final contribution in the dissertation addresses this by defining key cell and device criticality metrics. A sensitivity-aware standard cell layout optimization is demonstrated using the proposed criticality metrics.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Aggressive scaling of CMOS technology has created huge challenges for circuit analysis and optimization. Manufacturing tolerances in the process technology are not scaling at the same pace as the critical dimensions (CD) of a device due to process control limitations. As the ability to control critical device parameters is becoming increasingly difficult, it has resulted in significant variations in device length, doping concentrations, and oxide thicknesses. Two sources of variation continue to be the most important sources, namely effective channel length (L_{eff}) and threshold voltage (V_{th}). The optical lithography uses light sources with wavelength much larger than the minimum feature sizes for the technology. Therefore, controlling critical dimensions like L_{eff} at these technology nodes has become very difficult. Variations in V_{th} are primarily a result of variations in channel doping. Additionally, within-die variations due to the fundamental physical limits such as random dopant fluctuations (RDF) and line-edge roughness (LER) are increasing significantly with successive technology generations. In addition to the growing importance of within-die process variations, the total number of process parameters that exhibit variation and impact yield has also increased. [1] [2] [3] [4]

For over two decades, gate-level static timing analysis (STA) has been the industry standard for timing yield prediction and timing sign-off. Why is STA so successful and

pervasive? The primary reason is that it does not require identification of vectors to perform timing simulation and hence, can handle designs that have several millions of gates. Moreover, STA can be performed incrementally which allows it to be easy to use for all design synthesis and design optimizations. Another major advantage of the gate-level STA is that the method can use pre-characterized timing models for the cells/gates.

Process variations pose a significant problem for timing yield prediction. Traditionally, these process variations have been modeled using several timing corner models, where the timing corners are defined for specific frequency targets. The corner models generally account for global variations in the process parameters and a deterministic STA is then performed for each such corner model. Even though timing corners can capture the global variations fairly well, there is no good method for modeling the within-die random variations. Additionally, the number of timing corners that need to be defined to capture the growing number of variation parameters is increasing. This requires that the delay and the timing information in a design is now modeled not as a deterministic value but as a random variable.

Gate-level statistical static timing analysis (SSTA) performs timing analysis by modeling delay as a random variable. A statistical timing model for the gates/cells is pre-characterized (termed as ***statistical cell characterization***) and these models are then used within an SSTA tool to perform timing yield prediction. There has been significant research in this decade addressing several issues in SSTA. Most of the research have primarily addressed the challenges in building SSTA algorithms. However, from the perspective of getting the SSTA tools to be mature timing-sign off tools, efficient and scalable timing models need to be built that address both inter-die and within-die random variations.

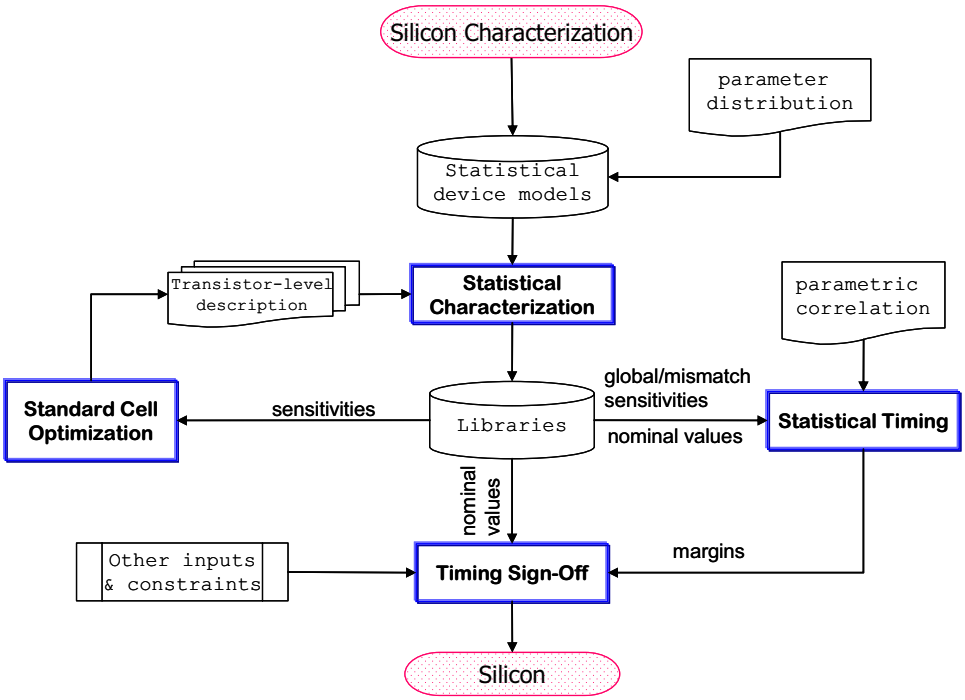


Figure 1.1: Overview of Statistical Characterization From Silicon to Design and Back to Silicon

This dissertation addresses the challenges of statistical timing characterization of digital cells and presents practical solutions and methodology to timing sign-off. Figure 1.1 illustrates the context of the contributions in this dissertation. An understanding of the sources of process variation starts from ***silicon characterization***. The results of silicon characterization are translated into ***statistical device models*** through a combination of physical modeling and empirical fitting. Transistor level circuit simulations and library characterizations are generally performed using the device models. The statistical gate delay models in the form of nominal delay and the delay sensitivities to different parameter variations are propagated into the design analysis and optimizations through ***statistical***

library characterization. A circuit design, now with an understanding of variations and enhancements for an improved timing yield goes back to silicon after a timing sign-off.

In the following sections an overview of the sources of process variations and silicon characterization are described. Challenges in statistical timing and state-of-the-art in this area are described in Section 1.4. A summary of the contributions in this dissertation and its organization are given in Section 1.6.

1.2 Sources of Process Variation

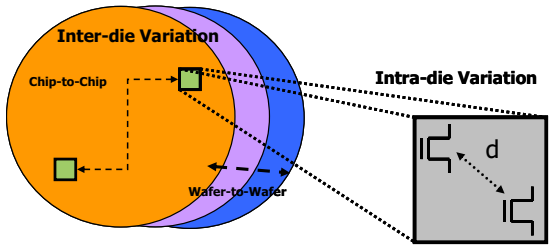


Figure 1.2: Classification of Process Induced Variations

Process variability in devices can be classified into two broad categories: (a) global variations and (b) local variations. Typically, all chip-to-chip, wafer-to-wafer and across-wafer variations are combined as global variations (also, commonly referred as inter-die variations). Variability across-chip (intra-die or within-die) is termed as local variation. Local variations can be further classified as geometry-dependent variations and mismatch variations. This is illustrated in Figure 1.2. [5]

1.2.1 Inter-Die Variations

Global or inter-die variations are due to unintentional shifts in the contemporary process conditions. It is typically not associated with fundamental physical limitations, but rather with the fabrication process. Global variability can be from lot to lot, from wafer to wafer within a lot, across wafers, or across reticles. The wafer processing steps that are sources of variation include a) rapid thermal anneal, when temperature gradients appear across the wafer, b) photo resist development, and c) etching. The primary source of variation for a reticle comes from the photo-lithography process. During photo-lithography, variability is caused if the focus changes as the mask is stepped across the wafer. Focus changes or variation can be caused due to lens aberrations in the exposure tool and/or due to wafer non-planarity. All of these variations translate to a die-to-die or inter-die variability. In an effort to achieve a robust manufacturing process, the inter-die variability is minimized by the process engineers as the process matures. [6] [7]

1.2.2 Within-Die Variations

Local/within-die random variations are caused by atomic-level differences between devices that occur even though the devices may have identical layout geometry and environment. These differences appear in dopant profiles, oxide thickness variation, and line-edge roughness. There are two significant contributors to within-die variations: (a) V_{th} variations and (b) Line-edge roughness (LER). The uncertainty caused by atomistic nature of dopant in MOS devices gives rise to significant V_{th} variations [2] [3]. These variations depend on the doping profile; generally, doping near the surface and close to the actual channel has the largest effect on V_{th}. LER arises from statistical variation during lithography exposure,

and depends on the absorption rate, chemical reactivity, and molecular composition of the photoresist [4]. Additionally, fluctuations in doping levels and device feature sizes also cause variation in the source/drain region, affecting the overlap capacitance and the effective source resistance. Even when the gate line-edge is perfectly smooth, the fluctuations in doping level cause uncertainty in the edge of the source and drain, which translates into source/drain capacitance and resistance variations. LER can exacerbate this effect.

In addition to the inter-die and within-die classification, process variability can be categorized to be either systematic or random variations based on the manner in which the variations are treated during the design cycle. A systematic variation may be modeled by estimating the impact of the variability on specific design style. Consider for example, the chemicalmechanical polishing (CMP)-induced relationship between the thicknesses of metal or inter-layer dielectric (ILD) and the layout feature density. It is possible to analyze the impact of the CMP process on a design and adjust the design layout and/or timing to mitigate the resulting variations. Random variations, however, require the designer to provide additional design margins that guards against performance loss.

Both random and systematic process variations reduce parametric yields significantly at the leading CMOS process technologies. Parametric yield, distinct from the traditional notion of defect-limited yield, reflects how variations in the physical characteristics alter the electrical characteristics. This in turn degrades yield due to variations in the circuit timing and power. These variations are stochastic in nature and cannot be described using purely deterministic physical models. The physical models need to be augmented with the empirical data coming from statistical characterization of silicon.

1.3 Silicon Characterization

As mentioned in the previous section, the inter-die and within-die process parameters exhibit systematic as well as random components of variations. Quantifying these components of variation and the resulting correlation among the parameters, requires a comprehensive set of characterization of real silicon. Such a process of silicon characterization (also commonly referred as statistical process characterization) allows for a better understanding of the sources of variations. Understanding the sources of variability can then help decrease the design margins and improve the competitiveness of a design.

Silicon characterization can be performed to study the "sources of variations" by directly looking for and measuring the variations that cause errors. It can be coupled with the study of sensitivities to each error source to predict the resulting error distributions. Consequently, there are two approaches to silicon characterization: (a) characterization of the input parameters; where a set of data from measured silicon estimates the error of the input parameters from the target values and then uses statistical approach to separate the errors into different components of variation and (b) characterization of the circuit output parameters; where a combination of measured statistics and physical models coupled with circuit variability simulation is used to characterize for different components of variation. These silicon characterizations that capture the statistics of the device or interconnect parameter variations as well as their electrical behavior complement the application of statistical analysis of the designs.[8][9]

An example we consider for characterization of input parameters is device CD using the lithography process. One of the most challenging problems in the lithography process today is to separate the systematic spatial components into their specific components for

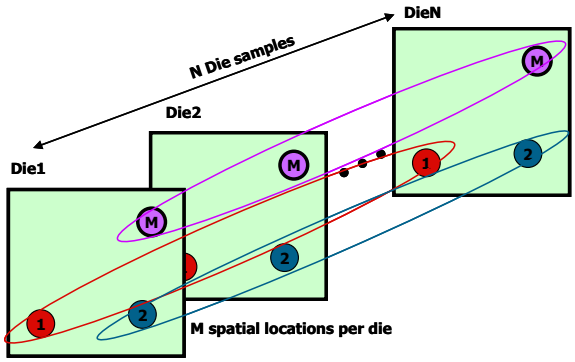


Figure 1.3: Characterization of Within-Die CD Variations

across-wafer, across-reticle and across-die. Consider for example, the statistical characterization of systematic spatial variations vs. random variations for intra-die variability. First measurement of a number of die at the exact same spatial locations is done (illustrated in Figure 1.3). Averaging of the measurements at each spatial location creates a composite die with the random errors mostly averaged out. Subtracting the composite die data from the measured die data produces a residual set of random errors that can be characterized by a mean and standard deviation. The composite die residual can then be used to characterize for the systematic spatial component of the error.

Even though there are sophisticated technology CAD tools that can simulate complex physical processes of individual manufacturing steps as well as electrical behavior, the core physical models employed often do not comprehend the variety of stochastic processes that contribute significantly to parametric yield. Hence, the direct measurement of device and interconnect variability, especially in the electrical parameters that impact the parametric yield, has become essential in the current process technologies. These measurements

complement the physical process and device models.

Silicon characterization requires creating test structures and test vehicles to measure statistics. During process development, the silicon characterization is performed using test vehicles that have dedicated mask sets. This provides a large area to embed thousands of different test structures for characterization of device performance, design rules and reliability optimizations. However, during production, only the scribe line is available along with the product masks for the test structures. Further, decomposition of the variations into systematic and random components require several structures to be replicated. This severely limits the number of test structures available for characterizations on production masks. New test structure strategies that allow for greater packing density in terms of area as well as allow for measurement of multiple statistics using the same circuitry need to be developed. [10]

1.4 Statistical Timing Analysis: State of The Art

Statistical timing analysis has attracted great attention in the past decade because it addresses several limitations of the corner-based approach of deterministic timing analysis [11] [12] [13] [14] [15] [16] [17] [18]. Conventional STA uses different timing corners assuming conservative process, voltage, and temperature (PVT) conditions which can yield an overly pessimistic analysis and leave valuable power and performance improvements on the table. The number of STA runs required can become intractably large when the designers attempt to reduce the pessimism in the analysis. Further, STA does not adequately account for within-die variations which can result in over-margining the designs.

SSTA models the circuit's performance as a random variable and accounts for the parameters

of variation by considering the parameters as distributions. The simplest SSTA approach is to perform Monte-Carlo (MC) analysis with multiple iterations of deterministic STA. In each such iteration, the distributions of all process parameters are sampled and a deterministic STA is run on the design for those values. The MC approach can handle all types of parameter distributions and parameter correlations. However, the approach requires a large number of deterministic STA iterations and becomes computationally expensive even for small circuits. More efficient SSTA algorithms have been proposed and are classified into two basic categories: namely, block-based analysis and path-based analysis.

1. ***Path-based Statistical Timing***: In a path-based approach, a deterministic STA is run on a design and the top few critical paths are selected. The delay of each path is then statistically analyzed using MC like simulations. This results in the probability distribution of each path delay, and a desired confidence point in the delay distribution is then compared with the target circuit's performance. The advantage of this approach is that it eliminates the problem of delay correlations due to path reconvergence. However, the number of paths falling within the desired confidence may be very large and the path-based approach can become expensive. The path-based approach also lacks the ability to perform incremental analyses which is imperative for circuit optimizations.

2. ***Block-based Statistical Timing***: Block-based statistical timing analyses are targeted at deriving the circuit's performance distribution to predict the timing yield. The delay of each gate is modeled as a random variable and the whole circuit's performance distribution is computed. This approach requires computing the first-order

sensitivities for all timing quantities of the circuit with respect to all sources of variation. The inherent need to compute circuit's performance sensitivity to different process parameters makes statistical analyses apt for circuit optimizations.

Due to the advantages in block-based SSTA, all research in the recent years have been in block-based SSTA. All references to SSTA in this dissertation will be for block-based SSTA. There are however several challenges in improving both accuracy and runtime of block-based statistical timing analysis and making it into a sign-off technology. Several of these challenges and the state-of-the-art in handling these challenges are described below.

Delay models: The first task of gate-level SSTA is to compute the statistical characteristics of cells/gates in the library for delay variation due to several process induced variations. There are very few works that have published models for statistical delay modeling. In [19] a model for gate delay variation was proposed and the dependence of delay variation on supply voltage was derived based on an alpha-power model. Recently, an analytical approach was used in [20] to develop a delay model to study the impact of random dopant variations on gate delay and in [21] an analytical delay model was developed to study the impact of process variations for the circuits functioning at sub-threshold and super-threshold regions. In all these works, the resulting models are restrictive to specific parameters and not scalable for different variation parameters that are used in the *SPICE* models. While analytical models provide good insight into impact of specific parameters on circuit performance, it cannot represent a scalable model across technology generations with increasing numbers of variations. In [22] the authors presented a generalized model that is based on the probabilistic collocation method [23] to construct statistical gate-delay models to different variation

parameters. The method requires sampling several points on the probability distribution of each parameter and hence results in large number of *SPICE* simulations. All these models represent the variation parameters at the cell level and do not model for the device-to-device fluctuations due to within-die variations.

MAX operation and Normality assumption: SSTA requires performing two atomic operations on random variables: (a) the SUM operation representing the sum of delays along the path and (b) the MAX operation representing the signal that is propagated from gate to the next in a timing graph. Typically all the variation parameters in SSTA are modeled as Gaussian random variables. The SUM operation on two Gaussian variables results in another Gaussian random variable; however, the result of MAX the operation is not a Gaussian variable. Most of the methods in the literature (e.g., [24] [14]) are based on modeling the delay as a linear function of Gaussian random variables. To maintain the output in the same form, the MAX operation on two delay variables (that are Gaussian) is approximated by another linear function of the same Gaussian random variables. Thus, the maximum of two Gaussian random variables is approximated by another Gaussian random variable whose mean and standard deviation are computed using Clark's approximations in [25]. Several techniques were proposed to improve the accuracy of using linear models and the non-Gaussian nature of the output variables. In [26] the authors describe a method to propagate linear delay models of non-Gaussian and Gaussian random variables. The max of two delays is also modeled as a linear function of random variables. A new MAX operation for skewed normal random variables as an alternative to Clarks approximation was proposed in [27]. Methods to propagate quadratic delay models [28], [29] have also recently been proposed. The MAX of two quadratic functions is approximated by a

linear function; the mean and variance of the MAX operator is determined by matching the computed moments with the moments of the approximated function and solving a set of linear equations. In all these models the sources/parameters of variation are assumed to have Gaussian distributions. To handle non-Gaussian parameter distributions, a numerical approach based on conditional probability combined with the first-order formulation of Gaussian distributions is proposed in [15].

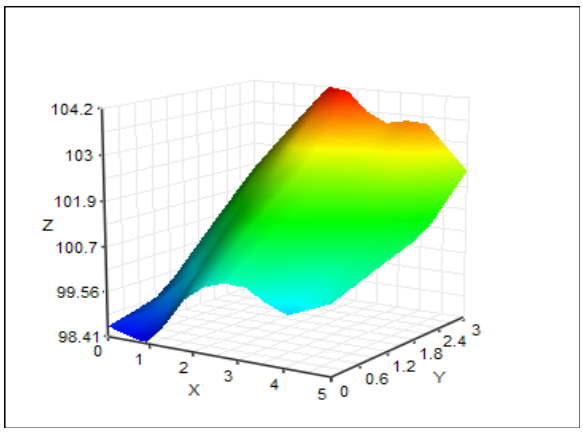


Figure 1.4: Spatial Signature for Device CD

Spatial Correlations: There are certain parameters of variation that exhibit systematic spatial within-die variations. An example is a systematic spatial signature for the device CD illustrated in Figure 1.4. Such a signature results in correlation between the timing variables of two gates within a design (termed as spatial correlation). There has been significant research that has gone into accounting for spatial correlations and their impact on timing[14] [30] [31] [32] [33] [34] [35]. Considering spatial correlations has significant impact in both modeling the sources of variation and the path criticality analysis. In [11] [14] [36] the authors use a grid-based approach to model spatial correlations. In the

grid-based approach, the circuit is partitioned into a grid number of cells such that the parameters of all the gates in a single cell are totally correlated. In [31] the authors handle spatial correlations by defining all parameters in the circuit using four principal components (corners of the circuit). In addition to spatial signature for process induced parameters, there are spatial variations due to voltage and temperatures variations within the chip.

Path Correlations: When two paths in a circuit share a common gate, net or path segment, it results in correlation between the path delays (termed as path correlation). The significance of path correlation comes when each gate has a certain local variation that is independent of other gates. When the local variations are propagated through the timing graph to the circuit's output, it causes additional correlations due to sharing of these local variations. Such correlations need to be accounted when computing the circuit's performance (maximum delay distribution). The authors in [37] propose a formulation to the first-order delay variations by including all the random variables due to each gate during timing analysis. In [16] a method based on common node detection is used to deal with path correlations. There are no good methods that handle the path-based correlations due to within-die mismatch variations. Additionally, path correlations arise during computation of clock-skew variations.

Input slew and output capacitance variations: The delay of a gate depends on a finite input transition time (input slew). Input slew has a strong impact on delay as well as delay variation. Input slew and capacitance load also vary with several parameters of variation. Accurate modeling of input slew and the load capacitance variations on gate delay and during timing propagation improves the accuracy of SSTA. Very little work in the

literature accounts for these variations. In [38] a numerical model is developed to account for input slew effect on delay variation. However, this model does not account for input slew variations due to within-die mismatch variations.

Statistical constraints: Sequential elements like flip-flops, registers and level-sensitive latches in the design impose a timing constraint on the data signals. For SSTA, these constraints need to be modeled as random variables. Characterizing the sequential elements for statistical constraints as well as accounting for these during timing propagation can be very challenging. For example, the analysis of flip-flop-based designs is a direct extension of the combinational circuits because the clock edge trigger the propagation of data from one pipeline stage to the next. However, handling level-sensitive latches in the design becomes very complicated because the paths between pipeline stages now become correlated. There are several research efforts that have focused on statistical analysis of latch-based designs [39] [40] [41] [42]. In [41] the problem is formulated as propagation of critical probabilities across pipeline stages. However, this method does not model for the cycle sharing between different stages.

Interconnect variations: Variations in interconnect geometry result in variations in resistance and capacitance of each metal layer in the design. Interconnect variations impact both the interconnect delay as well as the gate delay to which the interconnect network is connected. Several works have been published accounting for variations in the interconnect delay. In [43], the authors propose a model-order reduction framework to compute the effect of interconnect variations on delay. The authors in [44] [45] proposed statistical extensions to the closed-form interconnect delay metric. A statistical framework for modeling the effect

of crosstalk-induced delay noise was presented in [46] [47] [48]. In a recent work [48], the authors propose a closed-form expressions for modeling statistical delay noise that can be easily integrated into existing SSTA tools.

Process variations also impact silicon testing, diagnosis and debug that look to find the root-causes for design failures which are addressed poorly in the existing frameworks. There is very little work done in developing a methodology that looks for useful design information in the good/passing chip data. Also, there has been not much work done in systematically correlating the measured circuit delay distributions with the predicted distributions using SSTA methods and then correcting the statistical models using observed errors.

1.5 SSTA for Within-Die Random Variations

Within-die random variations are becoming the most dominant source of process variations. Study of delay mismatch between identical structures within-die shows that there is an increase in percentage of mismatch variations from one technology generation to the next. Figure 1.5 illustrates observations from silicon measurements on $65nm$ and $45nm$ test vehicles. For the same supply voltage the delay mismatch increases by as much as 25% when moving from 65nm technology to the 45nm technology. Further, there is a significant increase in the delay mismatch for smaller supply voltages. As we move from one technology to the next, the nominal supply voltage also reduces by a factor of $\sim 0.7X$. The reduced nominal supply will further exacerbate the effects of delay mismatch.

Gate-level statistical timing approaches lack good models for handling timing variations due to within-die random variations. During timing analysis, the global/inter-die variations increase linearly with increase in number of stages in a path; however delay vari-

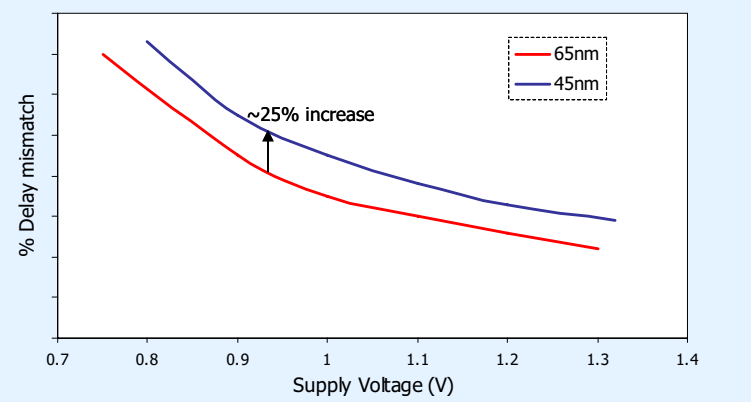


Figure 1.5: Delay Mismatch for $65nm$ and $45nm$ Technologies at Different Supply Voltages

ations due to the device mismatch variations increases as root-square-sum of stage delay variations. This is because the same inter-die parameter impact the cells on a path; however, within-die random variations result in device-to-device fluctuations resulting in large number of variation parameters that are statistically independent between devices and cells along the path. The relative standard deviation (sigma-over-mean) decreases along a path for mismatch variations. However, the mismatch variations has significant impact for timing constraints and for clock-skew variations (illustrated in Figure 1.6). Even for the most balanced circuits, the impact of mismatch variations is non-zero and significant on skew variations. It impacts both data-to-clock skew used to determine constraints and local clock-to-clock skew between flops. Timing constraints are typically determined as skew (time difference) between data and clock signals. For example, setup time constraint for a flip-flop is defined as the skew between data and clock signals. Similarly, local clock skew

is the skew between two clock signals arriving at the adjacent flip-flops. Mathematically, mismatch variations becomes significant when finding difference between two statistically identical and independent distributions.

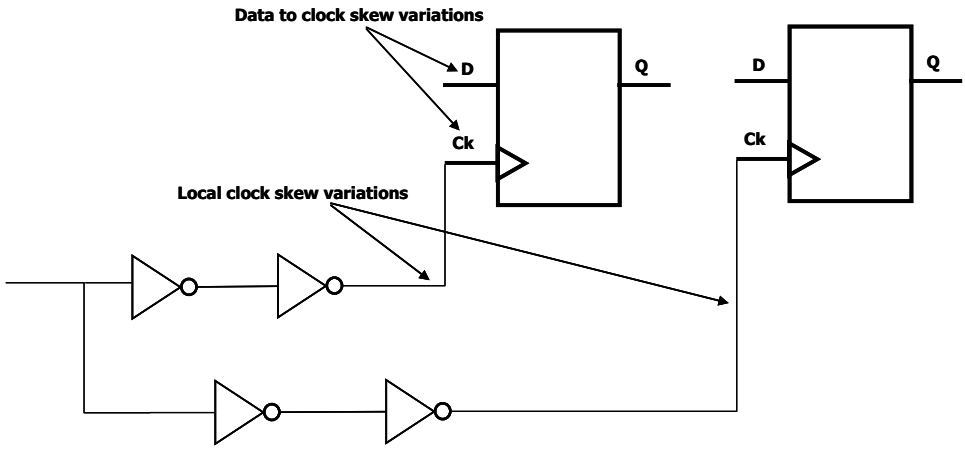


Figure 1.6: Impact of Mismatch Variations is Non-Zero and Significant on Skew Variations

It is challenging to both perform statistical characterization of cells for mismatch variations as well as performing statistical timing considering these mismatch variables. Even though these mismatch variables are statistically independent and uncorrelated, the mismatch delay variations can have correlations due to circuit structures like common devices shared between different timing paths, common path-segments or timing arcs shared between different paths etc. Additionally, since each device in the design represents a separate mismatch variable, the total number of variables in a design can easily become several millions. Handling such large number of variables during statistical timing can be computa-

tionally very expensive and intractable for large designs or SoC. The focus of this dissertation is to develop statistical methods for both gate delay models and timing analysis accounting for these within-die random variations.

1.6 Contributions and Organization of the Dissertation

An SSTA tool starts with presumed delay probability distributions for all gates/cells. Without an accurate statistical gate timing model, the SSTA tool will be incapable of precisely translating process variations into timing variability. While substantial efforts have been made to improve the efficacy of SSTA, the gate model which is the starting point has not been addressed sufficiently. In general, delay variability in gates can be characterized using rudimentary MC circuit simulations. MC simulations, though extremely valuable for verification and act as baseline for other techniques, are usually very expensive in computation.

In order to facilitate SSTA tools to be mature for timing sign-off, it is important to build delay models that are (a) simple to use in performance estimation and optimization tools, (b) scalable with increasing number of model parameters from one technology to the next and (c) able to provide insight into performance variations including variations in sequential cells and clock-skew variations due to the impact of both inter-die and within-die mismatch variations.

The centerpiece to a timing sign-off methodology is statistical library characterization (illustrated in Figure 1.1). The problem of statistical characterization can be simply defined as that of finding the sensitivity of delay, $D = D(\mathbf{P})$, with respect to several random parameters, $\mathbf{P}$. Sensitivity analysis is a well studied problem and several solutions exist

to finding the sensitivities. Review of the sensitivity analysis and variance decomposition methods are described in Chapter 2. Specifically, we highlight the analysis of significant factors using variance decomposition methods which is used for statistical characterization and optimization of the cells. We couple the knowledge of circuits along with the significant factor analysis methods to several problems including computation of the delay sensitivities, to perform statistical timing and to perform sensitivity-aware cell optimizations. The major contributions of this dissertation are described briefly below.

1. ***Statistical delay characterization***: Statistical characterization needs to be performed efficiently with acceptable accuracy as a function of several process and environmental parameter variations. In this work, we propose an approach to consider intra-cell process mismatch variations to characterize a cell's delay and output transition time (output slew) variations. A straightforward approach to address this problem is to model these mismatch variations by characterizing for each device fluctuation separately. However, the runtime complexity for such characterization becomes of the order of number of devices in the cell and the number of simulations required can easily become infeasible. The fluctuations in switching and non-switching devices and their impact on delay variations is analyzed. Using these properties of the devices, a clustering approach to characterize for cell's delay variations due to intra-cell mismatch variations is proposed. The proposed approach results in significant runtime improvement with acceptable accuracy, compared to Monte Carlo simulation results. It is shown that this approach ensures an upper-bound on the results while keeping the number of simulations for each cell independent of the number of devices. The proposed statistical delay characterization considering intra-cell mismatch variations

is described in Chapter 3.

2. ***Statistical constraint characterization***: In addition to statistical delay characterization of cells, the sequential cells like flip-flops and latches need to be characterized for setup and hold time constraints. The predominant computation time requirements during statistical library characterization are for constraint sensitivity computation. In Chapter 4, we propose a new delay-based approach for statistical characterization of constraint sensitivities. We show that the sensitivities obtained using such an approach can result in two-orders of runtime improvement comparing with traditional approaches, without much loss of accuracy.

3. ***Statistical timing considering correlations due to within-die mismatch variations***: As discussed earlier, advances in process technology have resulted in significant increase in within-die device mismatch variations. The mismatch variations are caused by variations or mismatch in device characteristics; as a result, fluctuations of each device in a cell impact the timing of the cell. Further, even though mismatch variations are considered random uncorrelated variations, during statistical timing analysis they can result in timing correlations. Additionally, block-based statistical timing approaches lack accurate consideration of impact of slew variations on delay and arrival time variations. This can result in significant errors in predicting circuit's performance distribution. In Chapter 5, we propose an approach to consider input-slew and gate-load variations using a common basis of parameter variations. We show that the proposed approach allows for handling correlations arising due to path reconvergence. Propagation of slew-based correlations in the timing graph results in explosion of variables. We present an efficient pruning technique to handle the large

number of variables without losing significant accuracy.

4. ***Timing margining considering within-die clock skew variations***: Timing margining is a key component of timing sign-off. Insufficient margin can lead to silicon failure and excessive pessimistic margin will entail unnecessary design optimization effort. Timing margin is intended to cover the uncertainty in clock arrival times and clock skews arising from within-die process variations. In highly scaled technologies, the increased process variations tend to enforce an overestimation of timing margins making it difficult for the designs to achieve the target performance. In Chapter 6, we present a more efficient margining methodology to account for clock-skew variations arising due to within-die variations. The proposed methodology fits well within current corner based timing sign-off framework and allows for significant reduction in margin pessimism. We present the results and observations on a low power processor for hold-time margin correction. Evaluation of the proposed methodology for hold analysis on a low power processor shows, on average, $\sim 67\%$ reduction in the original margin. Further the margin correction decreases the number of hold-time violations significantly and effectively achieves $10X$ reduction in effort to fix hold-time violations.

5. ***Sensitivity-aware standard cell optimization***: Standard cells are designed very early in the design cycle. Cells are designed before the process reaches production maturity level. At this early phases, accurate yield information may not be available. The cell optimizations generally do not account for any quantitative parametric yield information. However guidelines in terms of recommended design rules from prior knowledge of manufacturing issues are known. Any subtle improvements to reduce variability using these guidelines in standard cells can improve parametric yield sig-

nificantly. Reliable yield information may not be available until later in the process cycle. That requires, very subtle yield effects under variations to be estimated with good fidelity and used to guide standard cell designs before the process reaches volume production levels. We propose two new metrics namely *device criticality metric* and *total sensitivity index* that measure the significance of each device to standard cells under variations. We demonstrate that, using these metrics along with recommended layout design rules, the standard cells can be optimized for improved parametric yield with minimal or no penalty on the actual performance of the cells. The approach was implemented and results illustrated on a 45nm technology library. Chapter 7 presents the detailed formulations, results and challenges ahead.

In chapters 3 through 7, each chapter is organized into following sections: (a) an overview of the problem and challenges for an effective solution to such a problem, (b) the background information or a study of the problem, (c) a proposed solution to the problem, (d) experiments performed to validate the solution and finally, (e) a conclusion with challenges and issues that require future research focus.

Chapter 2

Sensitivity Analysis and Variance Methods

2.1 Overview

A typical adjunct to uncertainty analysis is sensitivity analysis, which attempts to determine how the uncertainty in individual elements of $\overline{x}$ affects the uncertainty in the elements of $\overline{y}$. To carry out the sensitivity and/or uncertainty analyses, the uncertainty in $\overline{x}$ must be characterized. For the discussions in this work, each i^{th} element of $\overline{x}$ is assumed to be characterized by Gaussian distribution, $N(\mu_i, \sigma_i)$.

Sensitivity analysis involves the determination of the effects of the individual elements of $\overline{x}$ on the function, $\overline{y} = f(\overline{x})$. Although sensitivity analysis is closely tied to the uncertainty analysis, it tends to be more complex due to both the variety of possible measures of sensitivity and the additional computational procedures required to evaluate sensitivity. One formal way to look at sensitivity analysis is to view it as an analysis of variance problem. Specifically, the variance, $V(\overline{y})$ of $\overline{y}$ is given by:

$$V(\overline{y}) = \int \left(E(\overline{y}) - f(\overline{x})\right)^2 \cdot g(\overline{x}).d\overline{x} \tag{2.1}$$

where $E(\overline{y})$ is the expected value of $\overline{y}$ and is given by following equation:

$$E(\overline{y}) = \int f(\overline{x}).g(\overline{x}) \cdot d\overline{x} \tag{2.2}$$

Sensitivity analysis can then be viewed as decomposition of $V(\overline{y})$ into components due to the individual elements of $\overline{x}$ with the value of these components actually giving the importance of each such component with respect to the total variance.

In the following, Section 2.2 gives an overview of useful properties of commonly used probability distribution, namely the Gaussian or Normal distribution. Section2.3 describes basic statistical inference methods including statistical estimation of normal distribution and hypothesis testing. Sensitivity analysis methods and variance decomposition are described in Section2.4. Background references for this chapter are [49][50][51][52][53][54][55][56].

2.2 Normal Distribution

The most widely used probability distributions is the Normal or Gaussian distribution. The distribution is named after Gauss who developed it using the fundamental results of Central Limit Theorem. The basic reason the Normal distribution works well is because of the observation from the Central Limit Theorem, which states that if the number of samples in a population is large, then the distribution of average of these samples has a Normal distribution. A random variable, X with Normal probability density function (referred to as PDF) is given as follows,

$$g(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/2\sigma^2} \tag{2.3}$$

where μ is the mean and σ^2 is the variance of the distribution. The Normal distribution is completely defined by these two parameters, μ and σ^2, also referred as scale and shape parameters of the distribution. The distribution is typically denoted as $N(\mu, \sigma)$. The

cumulative density function (CDF) is defined as the probability of the variable, X taking on all values less than x. That is

$$G(x) = P(X \leq x) = \int_{-\infty}^{x} g(x)dx \tag{2.4}$$

A random variable with $\mu = 0$ and $\sigma^2 = 1$ is called a standard Normal random variable. The notation $N(0, 1)$ is used to denote the standard Normal distribution. The CDF of a standard Normal variable, Z is given as: $\Phi(z) = P(Z \leq z)$. Since, the Normal distribution is a symmetric curve, $\Phi(-z) = 1 - \Phi(z)$. An important property to note is that if X is a Normal random variable, with $mean = \mu$ and $variance = \sigma^2$ and we can define $Z = \frac{X-\mu}{\sigma}$, then Z is a standard Normal variable with distribution, $N(0, 1)$. Creating a new random variable, Z using this transformation is referred to as *standardizing*.

In this thesis, if a reference is made for a 3σ probability then, it is referred to $P(\mu - 3\sigma \leq X \leq \mu + 3\sigma)$ or alternatively, $P(-3 \leq Z \leq +3)$. This probability is equal to: $2\Phi(3) - 1 = 0.9973$ that is, the 99.73 percentile of the CDF.

2.2.1 Correlation and Independence in Normal Distributions

Consider X and Y are two random variables with Normal distributions, $N(\mu_{_X}, \sigma_{_X})$ and $N(\mu_{_Y}, \sigma_{_Y})$ respectively. Let $\rho_{_{XY}}$ given by Equation(2.6) be the correlation factor for X and Y. The joint PDF of random variables, X and Y is given as

$$g_{_{XY}}(x, y) = \frac{1}{2\pi\sigma_{_X}\sigma_{_Y}\sqrt{1-\rho_{_{XY}}^2}} exp\left\{\frac{-1}{2(1-\rho_{_{XY}}^2)}[\frac{(x-\mu_{_X})^2}{\sigma_{_X}^2} - \frac{2\rho_{_{XY}}(x-\mu_{_X})(y-\mu_{_Y})}{\sigma_{_X}\sigma_{_Y}} + \frac{(y-\mu_{_Y})^2}{\sigma_{_Y}^2}]\right\} \tag{2.5}$$

where $\rho_{_{XY}}$ is the correlation between X and Y and is a dimensionless quantity that is used to compare the linear relationship between a pair of random variables. Formally, $\rho_{_{XY}}$ is

defined as

$$\rho_{XY} = \frac{E((X - \mu_X)(Y - \mu_Y))}{\sigma_X \sigma_Y} \tag{2.6}$$

A more detailed understanding of the correlation factors is described in Section2.5.

The two variables, X and Y are said to be independent if and only if the joint probability, $P(X \in R_X, Y \in R_Y)$ is equal to the product of individual probabilities, $P(X \in R_X)$ and $P(Y \in R_Y)$, where sets R_X, R_Y are in the range of X, Y respectively. That is

$$P(X \in R_X, Y \in R_Y) = P(X \in R_X).P(Y \in R_Y) \tag{2.7}$$

Consequently, it can be derived that if two variables are independent, then their correlation is zero. However, if the correlation factor is zero, then the two variables may not always be independent. Normal distributions have an elegant property: if $\rho_{XY} = 0$, then the joint PDF in Equation(2.5) can be re-written as

$$g_{XY}(x, y) = g(x).g(y) \tag{2.8}$$

The above relation says that the joint PDF of variables X and Y is equal to the product of individual PDFs. Equivalently, Equation(2.8) results in Equation(2.7). Hence, if variables X and Y are Normal distribution with correlation, $\rho_{XY} = 0$, then the variables are said to be independent. This is a useful property of Normal distributions that will be used often in this work.

2.3 Statistical Estimation

Statistical estimation and inference methods are a class of statistical methods where a sample data or population is available. Typically, we want to calculate a certain function

of sample observations which is then used in making inferences about the population. This function is termed as *statistic*. In the following section we give a background on statistical estimations [56].

Point Estimation is a method of estimating an unknown parameter for a given population. The estimator is a statistic that specifies how to use the samples to estimate unknown parameters of the population. Consider a population with mean, μ and variance σ^2. Let $X_1, X_2, ..., X_n$ be n samples of the population. An example of an estimator is the arithmetic average of the sample values. That is, $\widehat{\mu} = \sum_n (1/n) \cdot X_i$ is an estimator of μ.

Any estimator is considered to be unbiased if expected value of the estimator is equal to the actual value: $E(\widehat{\mu}) = \mu$. A bias, B is defined as the difference between the expected and the actual value of the estimator and is given as: $B = |\mu - E(\widehat{\mu})|$. The mean squared error in the estimator is then error introduced due to bias and the variance of the population distribution. That is,

$$\epsilon^2 = E[(\widehat{\mu} - \mu)^2] = B^2 + \sigma^2 \tag{2.9}$$

This is a very useful relation. The basis of variance decomposition and analysis of variance relies on computing this mean square error.

Confidence Intervals define an upper and a lower bound or limit for an estimator. These bounds define within which the actual value exists for a given confidence factor, $1 - \alpha$. For large samples, assuming a normal distribution, the bounds can be defined using the following probability relation,

$$P[-z_{\frac{\alpha}{2}} \leq Z \leq z_{\frac{\alpha}{2}}] = 1 - \alpha \tag{2.10}$$

where Z is a normalized variable. Consider a Monte-Carlo random sampling on the population with N samples. Let $\widehat{\mu}$ be the estimator of the sample mean μ. By choosing a confidence level, $1-\alpha$ and assuming a Gaussian distribution with known variance σ^2, the upper and lower bounds on μ is given as

$$\widehat{\mu} - z_{\frac{\alpha}{2}}\sigma/\sqrt{N} \leq \mu \leq \widehat{\mu} + z_{\frac{\alpha}{2}}\sigma/\sqrt{N} \tag{2.11}$$

From the number of samples that lie within the bounds a confidence factor, $\widehat{\alpha}$ can be computed for the population. If this factor $\widehat{\alpha}$ is larger than α, then the number of samples is increased till the α is greater than or equal to the computed confidence factor, $\widehat{\alpha}$.

Consider that the sample size, N is large such that it follows a normal distribution with a known or pre-determined variance of the population, σ^2. If the confidence factor is given ahead and the bias is specified, then the required sample size for the population can be estimated.

$$N = \frac{z_{\frac{\alpha}{2}}\sigma^2}{B} \tag{2.12}$$

This is a very useful metric that can be used for determining the sample size ahead. Consider for example the data is collected for the CD (critical dimension) of a $45nm$ technology process, from several lots for multiple wafers and multiple die-per-wafer. The target CD is given as $\mu_0 = 40nm$ with deviation, $\sigma = 4nm$. Now assume two components of variation are being studied, namely (a) across-wafer and (b) across-die. The question is what should be the number of wafers, N_w that need to be sampled and what should be the number of die N_d that need to be sampled? While the population distribution is the same, the bias specified for across-wafer, B_w is different from that for across-die B_d.

Generally, B_d is a systematic bias due to impact of lithography on different patterns on the die and is much tighter than B_w. Say, $B_d = \pm 1nm$ while $B_w = \pm 2nm$. If $\alpha = 5\%$, then $z_{\frac{\alpha}{2}} = 1.96$ and the minimum number of samples required are

$$N_w = (1.96 * 4/2)^2 \sim 15 \tag{2.13}$$

$$N_d = (1.96 * 4/1)^2 \sim 62 \tag{2.14}$$

This says that the number of die that need to be sampled across a wafer should be 15 and the number of samples within a die (at different locations on the die) should be 62 for specified bias conditions at 95% confidence level.

2.4 Sensitivity Analysis Methods

Consider a system with single response denoted as: $y = f(\overline{x})$. Sensitivity and variance analysis of y involves evaluation of effects of individual elements of $\overline{x}$ on y, while, uncertainty analysis involves evaluation of the cumulative distribution function (CDF) and/or complementary cumulative distribution functions (CCDF) of y. Both uncertainty and variance analysis rely on estimation methods for these evaluations. These techniques to perform variance/uncertainty analysis can be broadly classified into two categories, namely, (a) sampling methods and (b) fast probability integration methods. One class of these techniques is based on sampling the input parameter space and include methods like Monte-Carlo methods, Latin Hypercube sampling methods, etc. The other class of techniques provide a fast probability integration methods that include Differential analysis, Response Surface Method (RSM), Fourier Amplitude Sensitivity Test (FAST) etc. In the following sections we describe two methods namely, Monte-Carlo analysis and Differential analysis. We then extend the

Differential analysis formulations for variance decomposition, variance-based sensitivities and significant factor analysis. We define the following notations.

- x_i = the i^{th} element in $\overline{x}$
- N_s = the total number of samples for each element x_i
- x_i^j = the j^{th} sample of x_i for $j = 1, 2, ...N_s$
- $y_j = f(\overline{x}_j)$ and w^j is the weighting factor for y_j

2.4.1 Monte Carlo Analysis

One of the most popular methods of uncertainty/sensitivity analysis is use of Monte-Carlo methods. In Monte Carlo (MC) analysis, a random sampling from a possible range of the input parameters is performed followed by evaluation of the output for each such sample. Thus, in the MC analysis, a probabilistically based sampling procedure is used to develop a mapping from analysis inputs to the outputs. The advantage of this mapping is that it provides a basis for both evaluation of uncertainty (that is, evaluation of the CDF and CCDF) and sensitivity analysis (that is, evaluation of effects of individual elements of $\overline{x}$ on $y = f(\overline{x})$).

Based on the definition of the weighting factor, there are different sampling methods [50], (a) Random sampling (b) Latin-hypercube sampling (c) Stratified sampling. Once the samples are generated, evaluation of y creates a mapping from inputs to outputs: $\overline{x}_j \xrightarrow{\Re} y_j$ where $y_j = f(\overline{x}_j)$ for all samples, $j = 1, 2, ..., N_s$. Given this mapping between the input samples and the output, the expected value and variance of the output can be determined using the following estimates respectively:

$$\tilde{E}(y) = \sum_{j=1}^{N_s} y_j.w_j \tag{2.15}$$

$$\tilde{V}(y) = \sum_{j=1}^{N_s} \left[\tilde{E}(y) - y_j\right]^2 .w_j \tag{2.16}$$

In the above equations, w_j is the weighting factor which differentiates the sampling methods. The weighting factor generally depends on the sample size and a typical value is given as $w_j = 1/N_s$ for an equal probability across the input sample distribution. Monte-Carlo method works very well for any relation between the input space and the output, and can be applied to both linear and non-linear systems. The drawback of the method is that it requires a large number of samples to be generated and can become computationally expensive. An alternate to the sampling methods is a local sensitivity analysis method or fast probability integration methods. Differential analysis is one such method described in the following section.

2.4.2 Differential Analysis

The Differential analysis method is based on approximating the relation $y = f(x)$ using the Taylor series expansion. Using a first-order model, the expansion can be given as follows

$$\tilde{y}(\overline{x}) = f(\overline{x}_0) + \sum_i \frac{\partial f}{\partial x_i} \cdot (x_i - \mu_i) \tag{2.17}$$

where $\overline{x}_0$ is the expected value of $\overline{x}$, that is: $E(x_i) = \mu_i, \forall i$.

Once the approximation in Equation(2.17) is known, then the moments of distributions for y and hence, the CDF of y can be determined. Specifically, the expected value and variance are given as

$$E(y) \approx y_0 + \sum_i \frac{\partial f}{\partial x_i} \cdot E(x_i - \mu_i) = y_0 \tag{2.18}$$

$$V(y) = \sum_i (\frac{\partial f}{\partial x_i})^2 \cdot V(x_i) + 2 \cdot \sum_i \sum_{j=i+1} (\frac{\partial f}{\partial x_i}) \cdot (\frac{\partial f}{\partial x_j}) \cdot Cov(x_i, x_j) \tag{2.19}$$

where $Cov(x_i, x_j)$ is the covariance between terms x_i and x_j. Sensitivity analysis uses the partial derivatives, $\frac{\partial f}{\partial x_i}$ to determine the effects of individual elements, x_i on y. Thus, with the knowledge of the approximation in Equation(2.17), both uncertainty and sensitivity evaluations can be made. The most difficult part of differential analysis however, is to determine these partial derivatives. There has been significant research to develop techniques for determination of these derivatives including numerical techniques like finite-difference methods, adjoint sensitivity methods, etc.

Better approximations to y can be obtained using higher-order Taylor series expansions. For example, a second order approximation of the Taylor expansion is given in Equation(2.20).

$$\tilde{y}(\overline{x}) = f(\overline{x}_0) + \sum_i \frac{\partial f}{\partial x_i} \cdot (x_i - \mu_i) + 0.5 \cdot \sum_i \sum_j \frac{\partial^2 f}{\partial x_i \partial x_j} \cdot (x_i - \mu_i) \cdot (x_j - \mu_j) \tag{2.20}$$

The expected values and variance can be determined using this higher-order relation. However, including higher-order terms and correlations between the elements of $\overline{x}$ and the computation of $E(y)$ and $V(y)$ becomes very complicated.

2.5 Variance Decomposition

By grouping all components due to individual variables, x_i and into components due to interaction between individual variables, x_i and x_j, Equation(2.19) can be decomposed as follows [50].

$$V(y) = \sum_i V_i + \sum_i \sum_{j>i} V_{ij} + \cdots + V_{123\cdots N} \tag{2.21}$$

where V_i is part of the total variance $V(y)$ that has components including only terms x_i, V_{ij} is part of $V(y)$ that has components due to the interaction of elements x_i, x_j, V_{ijk} is part of $V(y)$ that has components due to the interaction of x_i, x_j, x_k, and so on, up to $V_{123\cdots N}$ that has interaction terms from all elements of $\overline{x}$. By normalizing Equation(2.21) with total variance, following sensitivity equation can be obtained.

$$1 = \sum_i S_i + \sum_i \sum_{j>i} S_{ij} + \cdots + S_{123\cdots N} \tag{2.22}$$

where each sensitivity component is given as follows

$$S_i = (\frac{\partial f}{\partial x_i})^2 \cdot \frac{V(x_i)}{V(y)} \tag{2.23}$$

$$S_{ij} = 2 \cdot (\frac{\partial f}{\partial x_i}) \cdot (\frac{\partial f}{\partial x_j}) \cdot \frac{Cov(x_i, x_j)}{V(y)} \tag{2.24}$$

$$\cdots so\ on \tag{2.25}$$

$$S_{123\cdots N} = \frac{V_{123\cdots N}}{V(y)} \tag{2.26}$$

Here, sensitivity, S_i is the fraction of total variance that is due to only the self-terms x_i, S_{ij} is fraction of $V(y)$ due to interaction between x_i, x_j and so on. Consider a case where

only self-terms and pairwise interactions need to be analyzed. Then, Equation(2.22) can be rewritten as

$$1 = \sum_i S_i + \sum_i \sum_{j>i} S_{ij} + R_{12} \tag{2.27}$$

If the case under discussion does not include any higher-order interactions, then $R_{12} \sim 0$. Further, if the interaction terms, $S_{ij} \sim 0$ (and, hence $R_{12} \sim 0$) then, it indicates that the correlation terms are insignificant. Alternatively, if $\sum_i S_i \sim 1$, then the contributions of each element x_i to y can be considered to be uncorrelated. An ordering of x_i based on the value of S_i can provide a first-order ranking of the importance of the elements x_i. This leads to the following discussion on analysis of the importance or significance factor of x_i.

2.5.1 Significance Factor Analysis

Sensitivity and variance analysis provides a mechanism to determine the impact or the effect of individual elements of $\overline{x}$ on y. For example, if a first-order Taylor expansion is used for the analysis, then the fractional contribution of x_i on y can be approximated using the partial variance component given as

$$V(y|x_i) \approx (\frac{\partial f}{\partial x_i})^2 \cdot \frac{V(x_i)}{V(y)} \tag{2.28}$$

where $V(y)$ is determined using Equation (2.19). Note that $V(y|x_i)$ is same as the sensitivity component, S_i, discussed in the previous section. An ordering of the components of $\overline{x}$ based on the fractional contributions, $V(y|x_i)$, provides a ranking of the importance on the basis of how much of the variance of y can be accounted by each element in $\overline{x}$.

Consider the elements in $\overline{x}$ to be independent to each other. Let μ_i be the nominal value for each element, x_i. Normalizing the first-order Taylor expansion in (2.17) with respect to the nominal or center-point, y_0, the gives following relation

$$\frac{y - y_0}{y_0} = \sum_i \frac{\partial f}{\partial x_i} \cdot \frac{\mu_i}{y_0} \cdot \frac{(x_i - \mu_i)}{\mu_i} \tag{2.29}$$

where y_0 is computed by setting each input element to its nominal value, $x_i = \mu_i$. By finding the variance of left-hand-side and right-hand-side of Equation(2.30) and considering that the elements in $\overline{x}$ are independent, the following relation can be derived:

$$\frac{V(y)}{{y_0}^2} = \sum_i (\frac{\partial f}{\partial x_i})^2 \cdot (\frac{\mu_i}{y_0})^2 \cdot \frac{V(x_i)}{{\mu_i}^2} \tag{2.30}$$

The advantage of the above relation based on normalization in Equation(2.29) is that it requires no additional information about the distribution of $\overline{x}$ and the elements of $\overline{x}$ can be ranked using only the partial derivatives.

If, however, additional information like the standard deviation of y is known and given as σ_y then the following normalization can be used to assess the importance of x_i.

$$\frac{y - y_0}{\sigma_y} = \sum_i \frac{\partial f}{\partial x_i} \cdot \frac{\sigma_{x_i}}{\sigma_y} \cdot \frac{(x_i - \mu_i)}{\sigma_{x_i}} \tag{2.31}$$

where σ_{x_i} is the standard deviation of each element x_i. Finding the variance of both left-hand-side and right-hand-side of Equation (2.31) gives the following.

$$1 = \sum_i (\frac{\partial f}{\partial x_i})^2 \cdot (\frac{\sigma_{x_i}}{\sigma_y})^2 \tag{2.32}$$

Note that in deriving both Equations (2.30) and (2.32), the elements in $\overline{x}$ are considered to be independent and hence, the covariance, $Cov(x_i, x_j = 0$.

2.5.2 Correlated Variables

There are two widely used correlation coefficients, namely, (a) Pearson Correlation Coefficients, ρ_{ij} given in Equation(2.33) and (b) Spearman Rank Correlation Coefficient, P_{ij} given in Equation(2.34).

$$\rho_{ij} = \frac{\sum_k (x_i^k - E(x_i)) \cdot (x_j^k - E(x_j))}{\sqrt{\sum_k (x_i^k - E(x_i))^2} \cdot \sqrt{\sum_k (x_j^k - E(x_j))^2}} \tag{2.33}$$

The correlation coefficient, ρ_{ij} provides a linear relation between two variables with the variables decreasing or increasing together. The value of ρ_{ij} consequently ranges from $[-1, 1]$. The absolute value of the correlation coefficient, $abs(\rho_{ij})$ results in a value between $[0, 1]$ where a value of zero indicates no correlation between the variables and a value of one indicates perfect correlation.

$$R_{ij} = \frac{\sum_k (R_i^k - \overline{R}_i) \cdot (R_j^k - \overline{R}_j)}{\sqrt{\sum_k (R_i^k - \overline{R}_i)^2} \cdot \sqrt{\sum_k (R_j^k - \overline{R}_j)^2}} \tag{2.34}$$

where R_{ik}, R_{jk} denote the rank of the k^{th} sample x_i^k, x_j^k, respectively, and $\overline{R}_i$, $\overline{R}_j$ denote the average of the ranks for all samples of x_i, x_j, respectively.

The rank correlations, R_{ij} quantitatively capture a subjective assessment of how the lower to higher values of one variable, x_i is associated with the lower to higher values of the other variable, x_j. The value of P_{ij} also ranges from $[-1, 1]$ and thus, provides a measure of the strength of the monotonic relationship between two variables. The advantage of rank correlations is that it is independent of any distribution information and hence can be applied to any type of sample.

Pearson's correlation coefficient provides a parametric statistic specifically when the distributions are Normal. If the distributions are not known or are not Normal, then non-parametric correlation coefficients like Spearman's Rank Correlation Coefficients are useful.

Chapter 3

Statistical Delay Characterization

3.1 Overview

Process disturbances are often described by device parameter variations which can be classified into two basic types: *global variations*, which are the same for all devices on a chip and *local variations*, which vary from device to device on the same chip. There are several SSTA techniques ([15][16][24][30][31][32][37][57][58]) proposed to account for both inter-chip (global) and intra-chip (local) variations. However, these techniques consider the variations at the cell level and do not account for intra-cell device-to-device mismatch variations. The delay variations of each cell accounting for intra-cell mismatch variations also need to be included in statistical timing analysis and hence, characterized during statistical characterizations.

In [38], a method to characterize the cells/gates in a library for delay and slew is presented. The method starts with modeling a nominal delay as a function of several parameters including process variables, supply, input-slew and output loading. This is done using a second order Chebyshev polynomial. In the presence of random variables corresponding to process variation parameters, the variables are normalized and the coefficients are obtained using Hermite polynomials as basis functions. The problem with such an approach is it requires large number of simulations to obtain the polynomials. The proposed approach

tries to handle intra-die correlations arising due to spatial correlations. The model however does not address the problem of intra-die mismatch variations which can be a significant component of cell delay variations. From a standard cell perspective, the intra-die mismatch variations result in fluctuations in each device within the cell and hence, delay variability due to each such component is termed as intra-cell delay variability.

A naïve and straightforward approach to computing intra-cell delay variability is to assign random variables for each device in the cell; such a model becomes infeasible when considering a large number of devices and a large number of mismatch variables. To address this problem, in [59], [60], a statistical gate-delay variation using response surface method is proposed. The model calculates intra-cell variability through sensitivity constants which are computed by considering the devices that lie only on the transition path (charging/discharging path). Even though the intra-cell delay variance for all the devices within the cell together is represented finally using a single statistic, computing the sensitivity constants requires an additional p characterizations (where p is the number of devices in the cell). In the worst-case, the run time complexity of characterizations for each cell will be $O(np)$, where n is the number of intra-die physical parameters.

In this chapter we present a new clustering-based approach to model intra-cell mismatch variations. This approach reduces the number of characterizations required to capture intra-cell mismatch variations significantly. We show that using this approach the run time complexity is $O(n)$. That is, the run time depends only on the number of intra-die parameters and is independent of the number of devices within the cell. Further, the approach ensures an upper-bound on delay variance which is desirable for timing analysis. Experiments indicate that the proposed approach models the delay variations due to intra-

cell mismatch within acceptable accuracy of Monte Carlo simulations. A major advantage of the proposed approach is that it needs little or almost no change to existing characterization infrastructure. Following are the specific contributions discussed in this chapter.

- We present a systematic study of the impact of intra-cell device fluctuations on delay variations. We define problem-specific sensitivity metrics using variance methods described in the previous chapters. These sensitivity metrics together define the significant contributors to intra-cell delay variations.
- We present a novel approach based on clustering multiple intra-cell variations to compute the delay sensitivity due to device mismatch variations.
- We show that the proposed approach has a computational complexity of $O(n)$, independent of the number of devices in the cell. Here, n = number of intra-die (local) variables.

The chapter is organized as follows. Section 3.2 provides a background on the global and local parameters of variation; the section also gives an overview of requirements for delay sensitivity characterization. Section 3.3 analyzes the impact of intra-cell mismatch variations on cell delay variability and derives the proposed approach. In Section 3.5, the proposed clustering method is described in detail. Experiments and accuracy analyses of several digital standard cells for delay variations are presented in section 3.6. The proposed method is illustrated for delay variation, but it can be easily extended for output slew variation.

3.2 Background

3.2.1 Global vs. Local Variations

Process variability in devices can be classified into two broad categories (a) global variations and (b) local variations. Typically, all chip-to-chip, across-wafer and wafer-to-wafer variations are combined as a global variation (also, commonly referred as inter-chip variation). Variability across-chip (intra-die) is termed as local variation. Figure 3.1 illustrates global and local variations.

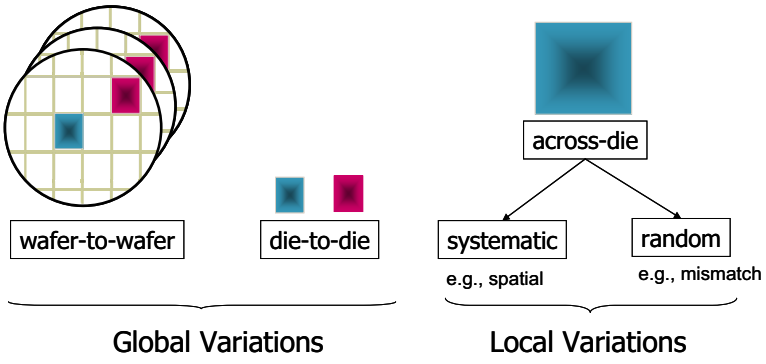


Figure 3.1: Process Variations: (a) Global Variations include Lot-to-lot, Wafer-to-wafer and Die-to-die Components (b) Local Variations include Within-die or Intra-die Components

Each parameter that has significant impact on the device characteristics can be represented in the following form

$$P = P_0 + \Delta P_g + \Delta P_l \tag{3.1}$$

where P_0 is the nominal or mean value, ΔP_g, ΔP_l is the global and local component of variations, respectively for this parameter. The components ΔP_g and ΔP_l are modeled as Gaussian random variables with distribution $N(0, \sigma_g)$ and $N(0, \sigma_l)$ respectively. Equivalently,

the random variables ΔP_g and ΔP_l can be modeled as $\sigma_g.\Delta X_g$ and $\sigma_l.\Delta R_l$ respectively, where ΔX_g and ΔR_l are normalized random variables.

To generalize for multiple parameters, let $\overline{\Delta X}$ be a vector of the global components of variation, $\{\Delta X_1, \Delta X_2, \ldots \Delta X_m\}$ and let $\overline{\Delta R}$ be a vector of the local components of variation, $\{\Delta R_1, \Delta R_2, \ldots \Delta R_n\}$. The global component ΔX_i varies from chip-to-chip; but, for a given chip this value is same for all devices in the design. The local component, ΔR_j is the across-chip component, which varies from device-to-device and captures both location or geometry dependent variations and mismatch or random variations. Vectors $\overline{\Delta X}$ and $\overline{\Delta R}$ are modeled as standard normal distributions, $N(0, 1)$ and are statistically independent of each other. The parameters within $\overline{\Delta X}$ (or $\overline{\Delta R}$) can be correlated in general. However, for simplicity of discussion we present the techniques below for uncorrelated parameters. If the parameters are correlated, an orthogonalization technique (for example principal component analysis) can be applied to extract uncorrelated parameters. For purposes of discussion in this chapter, any reference to local variations will be to only local-random variations (also termed as mismatch variations) unless specifically referenced. Local-random/mismatch variations are caused by variations or mismatch in device characteristics in a cell. In such case, fluctuations of each device in a cell impact the timing of the cell.

3.2.2 Statistical Characterization

Gate-level static timing analysis (STA) is a well known approach for timing sign-off. STA requires that the standard library cells are pre-characterized for delay and output transition time and stored in a two-dimensional table indexed by input transition time (input slew) and output load. Each gate/cell is characterized using a transistor level circuit simulator (e.g.,

Spice). Each cell in a library is characterized for m global parameters and n local parameters. Let p be the number of devices in a cell. From a cell characterization perspective, each variable in $\overline{\Delta X}$ impacts all devices identically and hence the delay variance represents a single statistic. However, each variable in $\overline{\Delta R}$ represents a separate random variable for each device in the cell illustrated in Figure 3.3. Since the magnitude of variations is much smaller than the nominal parameter value, usually performance characteristics like delay and slew of the cell is almost linear with respect to parameter variations. The basic idea is then to extract the first (mean) and second (variance) statistical moments of the performance metric (e.g., delay, output slew, etc.) and use them to represent the first-order statistical delay equation. The delay of a timing arc, D using a first-order Taylor expansion is given as follows,

$$D = D_0 + \sum_{i=1}^{m} d_i \Delta X_i + \sum_{j=1}^{n} \sum_{k=1}^{p} \sigma_{jk} \Delta R_{jk} \tag{3.2}$$

where D_0 is the nominal delay value, and is characterized by setting variations ΔX_i , ΔR_{jk} to zero. All ΔX_i, ΔR_{jk} parameters are modeled as $N(0,1)$. The quantities d_i and σ_{jk} are direct sensitivities of cell delay with respect to the global variations, ΔX_i and mismatch variations, ΔR_{jk} respectively. These are deterministic quantities obtained from characterization results. The problem of statistical characterization becomes one of determining these d_i and σ_{jk} quantities, which are delay sensitivities to the global and mismatch parameter variations.

Thus, characterization of cell-delay variation due to global variations is performed by varying a given parameter for all devices in a cell and due to mismatch variations, varying a given parameter for one-device-at-a-time. Typically, during statistical timing analysis, for each physical variable the mismatch components ΔR_{jk} for $k = 1 \ldots p$ are assumed to be

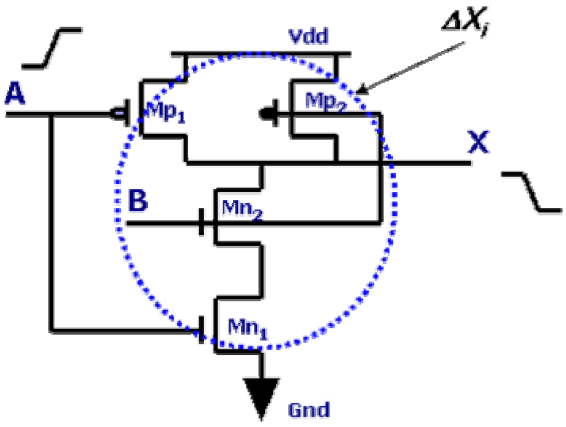


Figure 3.2: Global Variations in A Cell

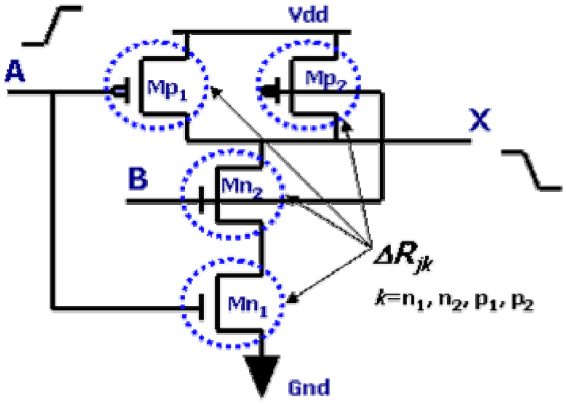


Figure 3.3: Mismatch Variations in A Cell

independent, uncorrelated variables. This enables the ΔR_{jk} variables for different devices to be combined to represent as a single variable. Thus, the above delay equation (3.2) can be rewritten as follows,

$$D = D_0 + \sum_{i=1}^{m} d_i \Delta X_i + \sum_{j=1}^{n} \sigma_j \Delta R_j \tag{3.3}$$

where σ_j is the equivalent delay sensitivity for parameters, $j = 1 \ldots n$ and it can be computed

using the relation: $\sigma_j = \sqrt{\sum_{k=1}^{p} \sigma_{jk}^2}$.

Consider the parameter of variation to be channel length, L with global variation component ΔL_g and local variation component, ΔL_l and is given as

$$L = L_0 + \Delta L_g + \Delta L_l \tag{3.4}$$

Figures 3.2 and 3.3 illustrate a 2-input NAND cell (NAND2) with four devices in two statistical characterization configurations: a. configuration for global variations and b. configuration for mismatch variations. Consider, delay variation for timing arc, $A(r) \rightarrow X(f)$ (input pin-A rising to output pin-X falling). To characterize for global variations, all devices are set to a single random variable, ΔL_g. The delay variation for each timing arc, $A(r) \rightarrow X(f)$ is determined with respect to this single parameter, ΔL_g. This is equivalent to all devices varying together in a correlated manner. For intra-cell mismatch variations, variations of each device in the cell impact the delay variation of timing arc $A(r) \rightarrow X(f)$. Figure 3.3 illustrates the configuration of a NAND2 cell with four devices for characterization with respect to the local variations, ΔL_l. In order to characterize for these local variations, each device i in the cell is assigned a separate random variable, ΔL_{li} and the effective delay variation due to all such intra-cell mismatch variables need to be determined.

Different approaches to characterize for delay variations due to local or intra-cell mismatch variations are described in the following section. Section 3.5 describes the proposed clustering-based approach.

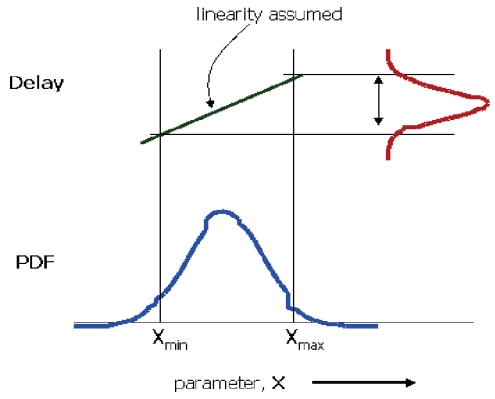


Figure 3.4: Characterization Using Numerical Difference: Linearity Assumed for Delay W.r.to Different Process Parameters

3.2.3 Finite Difference Method for Sensitivity Characterization

The process parameter variations are generally small comparing with the nominal parameter values. Within the range of parameter variation, delay is modeled as a linear function of the parameter values. Due to linearity, if the parameters are modeled with a Normal distribution, the delay follows a Normal distribution. This implies that given any two points X_{min} and X_{max} (see Figure 3.4), within the range of parameter variation, the delay can be given as follows,

$$D = D_0 + \frac{\partial D}{\partial X} \cdot (X - X_0) \tag{3.5}$$

Generalizing the above equation for several parameters results in the same equation as Equation 3.2. The variance of delay for a perturbation in X is given as

$$\sigma_D{}^2 = \{\frac{\partial D}{\partial X}\}^2 \cdot \sigma_X{}^2 \tag{3.6}$$

If X is $N(0,1)$, then $\sigma_X = 1$ and the delay sensitivity is $\frac{\partial D}{\partial X}$. This can be determined using a numerical difference method as follows,

$$\frac{\partial D}{\partial X} = \frac{D(X_{max}) - D(X_{min})}{X_{max} - X_{min}} \tag{3.7}$$

where $D(X_{min})$,$D(X_{max})$ are cell delay at parameter values, X_{min},X_{max} respectively. By normalizing all parameters and choosing to keep a single *nominal* condition for all parameters, the number of simulations to compute the delay sensitivity is $N + 1$, where N is the number of parameters of variation.

3.3 Modeling of intra-cell variations

3.3.1 Intra-cell variations

Consider a 2-input NOR (NOR2) cell as illustrated in Figure 3.5 for analysis of intra-cell delay variations; the number of devices, $p = 4$ for this cell. Let K be a physical parameter exhibiting mismatch variations (e.g., channel length with mismatch component, ΔL_l). Let ΔK_{N_i} and ΔK_{P_j} be the random variables corresponding to K for each nMOS device, N_i, and pMOS device, P_j, in the cell respectively. Let $\sigma_{K_{N_i}}$, $\sigma_{K_{P_j}}$ be delay sensitivities due to these random variables, ΔK_{N_i}, ΔK_{P_j} respectively. The problem of statistical delay characterization for intra-cell mismatch variations is then to determine cell's delay sensitivity, σ_K, as a function of $\sigma_{K_{N_i}}$ and $\sigma_{K_{P_j}}$. The most accurate method to find delay sensitivities is to perform a Monte Carlo simulation. In this case, the nMOS and pMOS parameters ΔK_{N_i} and ΔK_{P_j} respectively are varied randomly and the resultant delay variation is measured. However, Monte Carlo simulation will be prohibitively expensive. Monte Carlo simulation results are used as baseline for all accuracy comparisons.

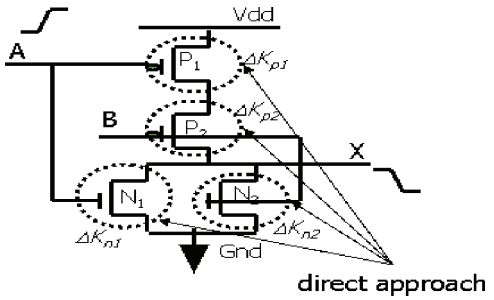


Figure 3.5: Direct and Simple Approach for Characterization of Local Variations

3.3.1.1 Simple Approach

A direct and simple approach to computing intra-cell delay variation of each timing arc is to determine delay variation by considering a random fluctuation in each device separately. Each variance component $\sigma^2_{K_{N_i}}$, $\sigma^2_{K_{P_j}}$ (for devices, N_i, P_j respectively) can be obtained through a separate characterization by setting random variables, ΔK_{N_i} , ΔK_{P_j} (illustrated in Figure 3.5). Assuming delay variation due to each device is statistically independent, the cell's delay sensitivity for each parameter K can be obtained using following relation:

$$\sigma^2_K = \sum_i \sigma^2_{K_{N_i}} + \sum_j \sigma^2_{K_{P_j}} \tag{3.8}$$

Thus, for a 2-input NOR cell with *four* devices, using the finite difference method described in Section 3.2.3 requires at least *five* simulations to determine the four mismatch components, $\sigma_{K_{N_i}}$, $\sigma_{K_{P_j}}$. If there are p devices in a cell then, at least *p+1* (one additional for the nominal value) simulations need to be performed to determine the effective mismatch sensitivity, σ_K. For n local sources of variations, the order of computational complexity is *O(np)*. While this approach is fairly accurate, it depends on the number of devices in the cell. This is not a good approach because, if the number of devices, p in a cell increases, the problem can become infeasible. Note that due to common device terminals and parasitics present for

each component of delay variance computation, there is correlation between delay variances, $\sigma^2_{K_{N_i}}$ and $\sigma^2_{K_{P_j}}$. However, this correlation is not significant and for all practical purposes can be ignored for digital cells (this is demonstrated empirically in Section 3.4).

3.3.1.2 Transition-path based Approach

Another approach to the problem of computing σ_K is to consider only devices on the transition path [60]. Each output transition is a result of conduction through a set of devices in the path from output to the power/ground rail (also termed as conducting path or transition path). For example, consider the $A(f) \rightarrow X(r)$ timing arc of the NOR2 cell (as illustrated in Figure 3.6). The devices P_1 and P_2 lie on the transition path from Vdd rail to output X. The delay sensitivity, σ_K, for $A(f) \rightarrow X(r)$ using transition-path based fluctuations is then given as

$$\sigma^2_K = \sigma^2_{K_{P1}} + \sigma^2_{K_{P2}} \tag{3.9}$$

This approach assumes that, σ_K has major contributions from P_1, P_2 and devices N_1, N_2 that are not on transition path are not significant contributors to the delay variation. This approach has the advantage that the *number of variables* considered for characterization of delay variance of each timing arc is reduced. For example, the NOR2 cell has at most 2-devices on the conducting path. However, it can be quickly observed that, using this approach, the number of devices that need to be considered for each delay variance computation is different from one cell to another and different from one timing arc to another. For example, in the case of the NOR2 cell, there are four timing arcs: $\{A(f) \rightarrow X(r)$, $A(r) \rightarrow X(f)$, $B(f) \rightarrow X(r)$, $B(r) \rightarrow X(f)\}$ with devices $\{(P_1, P_2), (N_1), (P_1, P_2), (N_2)\}$ that need to be identified respectively for each transition path. Effectively, the number of random variables that need to be considered for characterization is equal to the number of

devices in the cell. Further, this approach ignores contribution(s) from the switching device that is not on the transition path. That is, for a falling (rising) transition on the output, the pMOS (nMOS) devices are not on the transition path, and this approach considers these devices do not contribute to the equivalent delay sensitivity. We show in the following sections that this is not the case and all switching devices have significant impact on the timing arc delay variance. We use statistical methods and sensitivity analysis to prove these concepts. Further, we derive an approach that guarantees an upper bound on the cell's delay sensitivity/variance.

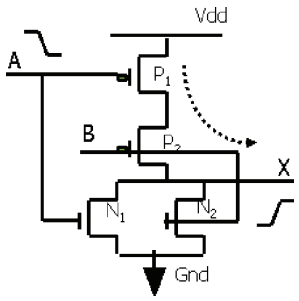


Figure 3.6: Devices on Transition Path = $\{P_1, P_2\}$ for $A(r) \rightarrow X(f)$

3.4 Study of intra-cell delay variations

The objective of this study is twofold: (a) first is to determine if the correlations between the intra-cell mismatch variations is statistically significant and (b) second to identify what are the significant contributors to effective mismatch sensitivity. We performed variance analysis to study the impact of intra-cell mismatch variations on cell delay variance. We use the sensitivity analysis methods described in Chapter 2 and propose problem-specific sensitivity metrics to analyze for significant contributors to the cell's effective mismatch sensitivities.

Several Monte Carlo simulations were performed by setting ***one device*** fluctuation at a time and by setting ***all*** devices randomly. The delay variance obtained by treating all devices randomly is treated as baseline. The Monte Carlo settings are explained in detail below

- ***Case 1*:** Monte-Carlo simulations were performed by treating ***all devices*** randomly and the delay variance or standard deviations for each timing arc are captured. For a given local parameter there is one Monte Carlo simulation performed in this case. Let σ_{all} be the delay sensitivity obtained, which forms the baseline simulation results.

- ***Case 2*:** Separate Monte-Carlo simulations were performed by setting ***one device*** fluctuation at a time and keeping the other devices at nominal conditions. For each local parameter, the number of Monte Carlo simulations in this case is equal to the number of devices. For example, for a NOR2 cell and for a single local parameter, say channel length, there are four separate Monte Carlo simulations performed once for each device. Let the delay standard deviations be: σ_{N_i} [1] for i^{th} nMOS device and σ_{P_j} for j^{th} pMOS device.

We define an equivalent delay sensitivity, σ_{eq} using variations from ***Case 2*** as follows,

$$\sigma_{eq}^2 = \sum_i \sigma_{N_i}^2 + \sum_j \sigma_{P_j}^2 \tag{3.10}$$

Monte Carlo simulation results for a NOR2 cell are illustrated in Figure 3.7 for input pin A transitioning and Figure 3.8 for B transitioning. Each bar in these charts depict six

[1] For simplifying notations, all subscripts for parameter K on delay sensitivity are dropped from this section onwards

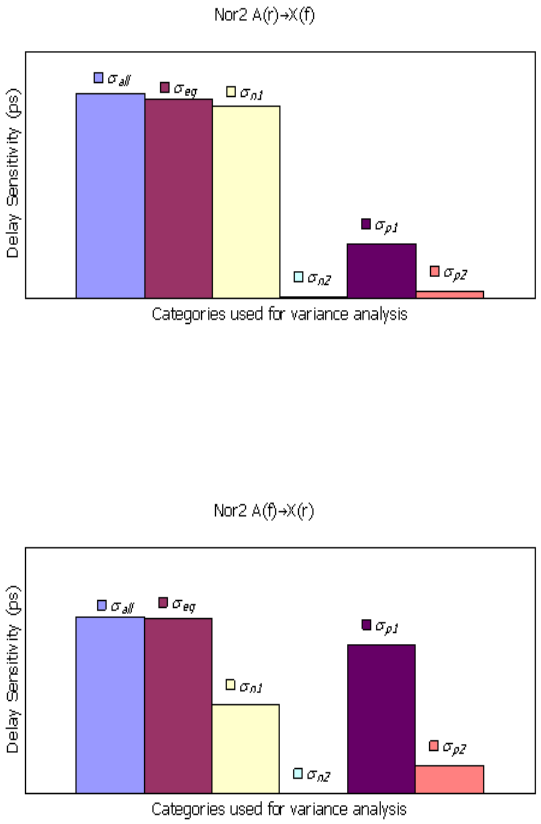


Figure 3.7: NOR2 $A \rightarrow X$: Impact of Individual Device Fluctuations Compared With Baseline, σ_{all} and Equivalent, σ_{eq}

results namely: σ_{all} from ***Case 1***, σ_{eq}, σ_{N_1}, σ_{N_2}, σ_{P_1}, σ_{P_2} from ***Case 2***. The results σ_{N_1}, σ_{N_2}, σ_{P_1}, σ_{P_2} are obtained by setting only one device fluctuation, N_1, N_2, P_1, and P_2 respectively. When input A (B) is transitioning, the devices N_1, P_1 (N_2, P_2) connected to these inputs at the gate terminal are defined as switching devices. The remaining devices are termed as non-switching devices. From the results, it is observed that the sum of delay

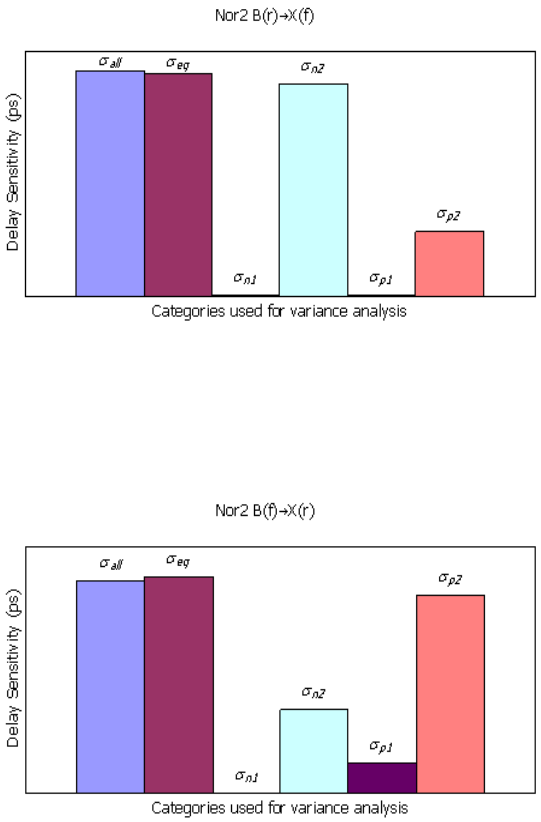


Figure 3.8: NOR2 $B \rightarrow X$: Impact of Individual Device Fluctuations Compared With Baseline, σ_{all} and Equivalent, σ_{eq}

variances obtained for each device fluctuation from ***Case 2***, σ^2_{eq} is almost equal to the delay variance from ***Case 1***, σ^2_{all}. That is, for all timing arcs, the ratio $\frac{\sigma_{eq}}{\sigma_{all}} > 0.98$. The χ^2 statistic for each timing arc are given in the following Table 3.1.

From the statistic in the Table 3.1, it can be observed that for all timing arcs, the ratio $\frac{\sigma_{eq}}{\sigma_{all}} > 0.98$, indicating strongly that the delay variance due to each device fluctuation

Table 3.1: $\chi^2 = \sigma_{eq}^2/\sigma_{all}^2$ Statistic for Each Timing Arc in the NOR2 Cell

Timing Arc	$B(r) \rightarrow X(f)$	$A(r) \rightarrow X(f)$	$B(f) \rightarrow X(r)$	$A(f) \rightarrow X(r)$
χ^2	0.974	0.952	1.045	0.989

can be considered to be statistically independent.

3.4.1 Significant Contributors

Let us revisit the concept of variance decomposition and significance factor analysis based on first-order Taylor series expansion from Sections 2.5 and 2.5.1. Then delay equation for mismatch variations can be rewritten as

$$\frac{D - D_0}{\sigma_{all}} = \sum_j \sum_k \frac{\sigma_{jk}}{\sigma_{all}} \cdot \Delta R_{jk} \tag{3.11}$$

Finding variance of above equation gives the following relation:

$$1 = \sum_j \sum_k \frac{\sigma_{jk}^2}{\sigma_{all}^2} \tag{3.12}$$

It can be observed that the ordering of the ratio, $\frac{\sigma_{jk}^2}{\sigma_{all}^2}$ provides ranking of the devices in terms of its contribution to delay variations. We use these analysis techniques coupled with the information that a CMOS gate is decomposed into clusters of pMOS and nMOS devices and define problem-specific sensitivity metrics that simplify the analysis of significant contributors for CMOS circuits.

First, we define a cluster as a set of devices of same type (either nMOS or pMOS). Any CMOS gate can be grouped into a set of nMOS devices (pull-down chain) and a set of pMOS

devices (pull-up chain). Now, define the variance components for each cluster as:

$$\sigma^2_{n-cluster} = \sum_i \sigma^2_{N_i} \tag{3.13}$$

$$\sigma^2_{p-cluster} = \sum_j \sigma^2_{P_j} \tag{3.14}$$

In order to determine the significant components (or contributors) of delay variance for each timing arc, we define two types of sensitivity.

- ***Direct sensitivity***: is ratio of delay variation determined from ***Case 2*** for each device fluctuation with respect to that determined using ***Case 1***. This can be represented for nMOS devices as $S_{N_i} = \frac{\sigma_{N_i}}{\sigma_{all}}$
- ***Cluster sensitivity***: is ratio of delay variation due to each device fluctuation with respect to variation of all devices in a single cluster. This sensitivity is computed for nMOS devices as $C_{N_i} = \frac{\sigma_{N_i}}{\sigma_{n-cluster}}$

The above sensitivity relations can be similarly determined for pMOS devices. From the previous sub-section, $\sigma_{eq} \approx \sigma_{all}$. Hence, the sum-of-squares of direct sensitivities for all devices is ≈ 1. That is, $\sum_i S^2_{N_i} + S^2_{Pi} \approx 1$. Thus, the ordering of devices based on direct sensitivities gives a measure of the significant contributors to timing arc's overall/effective delay variation.

An important property of the cluster sensitivity is that, the sum-of-squares of the sensitivities for all devices within the cluster is 1. That is, $\sum_i C^2_{N_i} = 1$ and $\sum_i C^2_{Pi} = 1$. Thus, ordering of cluster sensitivity provides information about which device fluctuation is significant contributor within a given cluster. Figures 3.9 and 3.10 illustrate the direct

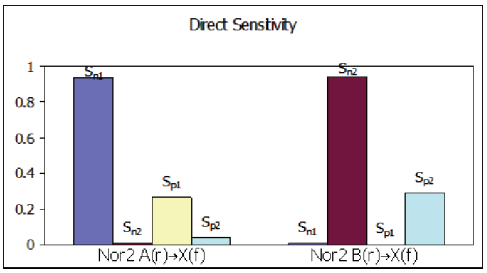


Figure 3.9: Direct Sensitivities NOR2 $A(r), B(r) \rightarrow X(f)$

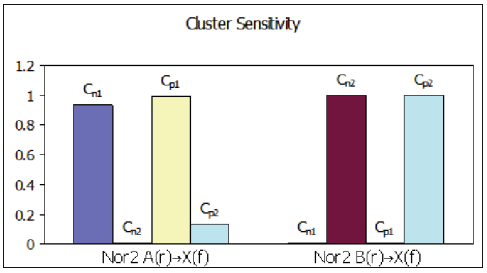


Figure 3.10: Cluster Sensitivities NOR2 $A(r), B(r) \rightarrow X(f)$

and cluster sensitivities, respectively, computed for each device of the NOR2 cell. The observations from the analysis of these results are described in the following sub-sections.

3.4.2 Delay sensitivity to switching device fluctuations

Statistical characterization is generally performed by considering one input pin in a cell to be switching at a time. For a typical CMOS gate, each such switching input is connected to atleast one nMOS and one pMOS device. These devices are termed as switching devices. The switching device on transition path results in significant portion of baseline variations, σ_{all}. For a falling output transition, a switching nMOS device is the primary contributor while, primary contributor for a rising transition is a switching pMOS device. Additionally, impact of switching devices that are not on the transition path is not insignificant.

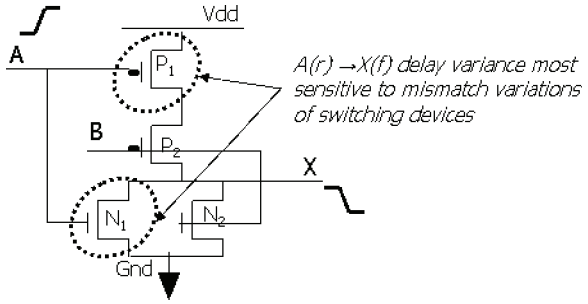


Figure 3.11: Impact of Switching Device Fluctuations

Consider results for $A(r) \rightarrow X(f)$ timing arc illustrated in Figure 3.7. In this case, the device on the transition path is N_1. The switching devices are N_1, P_1. The combined delay variance due to these switching devices comprise greater than 96% of total delay variance obtained from ***Case 1*** baseline results. That is, $S^2_{N1} + S^2_{P1} > 0.96$. Direct sensitivity analysis shows that impact of N_1 device fluctuation has the largest sensitivity (see Figure 3.9). Cluster sensitivity analysis for the same timing arc in Figure 3.10 illustrates that switching device fluctuations within a cluster comprise of greater than 98% sensitivity. That is, for timing arc $A(r) \rightarrow X(f)$, $C^2_{N1} > 0.98$ and $C^2_{P1} > 0.98$. Hence, switching devices are the most significant contributors.

3.4.3 Delay sensitivity to non-switching device fluctuations

Devices that are not connected to the switching input pin are termed as non-switching devices. Impact of non-switching devices, whether on transition path or not, is negligible.

For example, consider timing arc, $A(r) \rightarrow X(f)$. In this case, the non-switching devices are N_2 and P_2. The nMOS device, N_2 is not on transition path; and, P_2 lies on the transition path. Both N_2 and P_2 exhibit very small direct and cluster sensitivities and their contributions to the delay variations can be ignored. That is, $C_{N1} >> C_{N2}$ within the nMOS

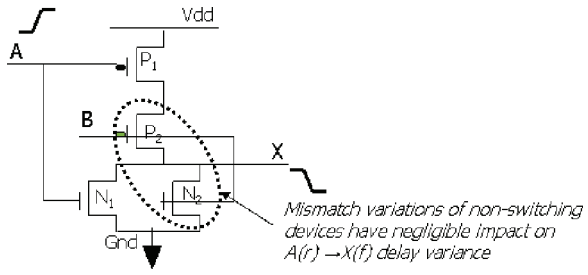


Figure 3.12: Impact of Non-Switching Device Fluctuations

cluster; and $C_{P1} >> C_{P2}$ within the pMOS cluster. When comparing the contributions for switching devices within a cluster, the impact due to non-switching device fluctuations is very small and can be neglected for practical purposes.

3.4.4 Intra-cell delay correlations

For each output transition, there can be correlation in delay between the timing arcs due to common device fluctuations. However, using direct and cluster sensitivities it can be trivially shown that such correlation is negligible. From previous Sections 3.4.2 and 3.4.3, it was observed that each timing arc has only its switching devices to be significant contributors and all non-switching devices can be ignored for practical purposes. As a corollary to this observation, the switching devices for one timing arc does not overlap with switching devices for another timing arc in the same cell. That is, the set of significant contributors for a given timing arc does not overlap with that for other timing arcs in a given cell.

Consider an example for the NOR2 cell. If output pin X is a falling transition, then, $A(r) \rightarrow X(f)$ and $B(r) \rightarrow X(f)$ can exhibit correlations because of common devices. However, the set of significant contributors $\{N_1, P_1\}$ for $A(r) \rightarrow X(f)$ does not overlap with the set of significant contributors $\{N_2, P_2\}$ for $B(r) \rightarrow X(f)$. Thus, if non-switching

device fluctuation impact is neglected, there are no common devices that contribute to the delay variance between $A(r) \rightarrow X(f)$ and $B(r) \rightarrow X(f)$, resulting in negligible correlation between $A(r) \rightarrow X(f)$ and $B(r) \rightarrow X(f)$ delay variations. Note that this is the case only for mismatch/local-random variations, and such correlation between timing arcs cannot be neglected when considering global variations and/or spatial-dependent variations in parameters.

3.5 Proposed Approach: Clustering-based intra-cell variability

The proposed approach takes advantage of the observations made in the previous section. The following properties are derived from analysis in Sections 3.4.2 and 3.4.3.

Property I: Impact of variations in switching devices both on the transition path and on the non-transition path form significant contributors to intra-cell delay variations.

Property II: Impact of variations in non-switching devices is small and can be negligible.

We take advantage of these two important properties and propose a new approach to characterizing for intra-cell delay variations as follows. The basic idea of the approach is to group all devices on the nMOS and pMOS stack separately, resulting in two clusters for each cell. Then assign fluctuations or random variables to the cluster instead of each device. This is equivalent to mapping any combinational cell to an inverter-like structure (see Figure 3.13). Since a cell is characterized for one input switching at a time, each cluster then has one switching device for a given timing arc. Within a cluster, the delay variance is most sensitive to the switching device and the non-switching devices have negligible contribution. As a result, the delay variations computed for the cluster random variable is same as that for the switching device. The delay variations thus derived for nMOS and pMOS clusters are

statistically combined to give the cell's equivalent delay variations due to intra-cell mismatch variations. This cluster-based approach is explained in detail below.

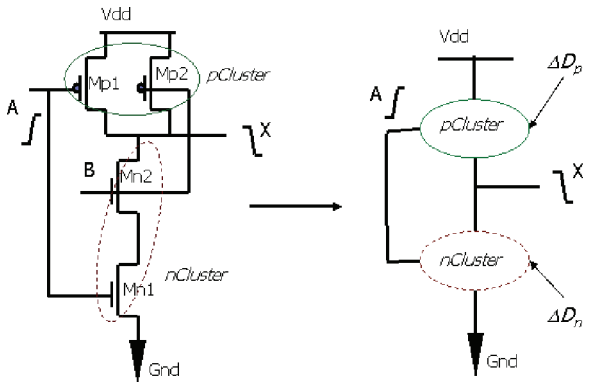


Figure 3.13: Clustering of nMOS/pMOS Stack: Equivalent to An Inverter With Two Delay Variables

Let ΔK_{N_i} and ΔK_{P_j} be random variables corresponding to physical parameter K for each nMOS device, N_i, and pMOS device, P_j, in the cell respectively. Now, group all nMOS devices into n-cluster and pMOS devices into p-cluster. Assign random variables, ΔK_n (ΔK_p) to the n- (p-) cluster corresponding to K. That means, every device in the nMOS (pMOS) cluster is assigned the same random variable, ΔK_n (ΔK_p). Consider ΔK_n (ΔK_p) to be standard normal distributions, $N(0,1)$. Let ΔD_n (ΔD_p) be the cell's delay variables due to ΔK_n (ΔK_p) as illustrated in Figure 3.13. Assuming linearity, ΔD_n and ΔD_p are also Gaussian with distributions, $N(0,\sigma_n)$ and $N(0,\sigma_p)$ respectively. Since all nMOS device fluctuations are varied together for the cluster, the delay sensitivity for n-cluster is

$$\sigma_n = \sum_i \frac{\partial D}{\partial K_{N_i}} . \Delta K_{N_i} \tag{3.15}$$

Since, ΔK_{N_i} are i.i.d. and $N(0,1)$, Equation (3.15) can be rewritten as

$$\sigma_n = \sum_i \sigma_{N_i} \tag{3.16}$$

where σ_{N_i} is delay sensitivity due to ΔK_{N_i}. Similarly, the delay sensitivity for p-cluster is

$$\sigma_p = \sum_j \sigma_{P_j} \tag{3.17}$$

Consider each device to be single fingered (handling of multi-fingers is explained in the following sub-section). For single input switching, there is a single nMOS and single pMOS switching device in a typical CMOS combinational cell. Let the index be $i = 1$ ($j = 1$) for the nMOS (pMOS) switching device in the n-(p-) cluster. All other devices within the cluster are non-switching. Dividing equation (3.16) with equivalent delay sensitivity for n-cluster, it can be rewritten as

$$\frac{\sigma_n}{\sqrt{\sum_i \sigma_{N_i}^2}} = C_{N1} + \sum_{i, i \neq 1} C_{N_i} \tag{3.18}$$

Using the cluster-sensitivity analysis from Section3.4, the cluster sensitivity of switching device, C_{N1} is significantly larger than cluster sensitivity of non-switching devices; hence, the last term in the above equation is negligible. Equation (3.18) can be rewritten as

$$\frac{\sigma_n}{\sqrt{\sum_i \sigma_{N_i}^2}} \approx C_{N1} \Rightarrow \sigma_n \cong \sigma_{N1} \tag{3.19}$$

Similarly, using cluster sensitivity analysis, the delay sensitivity of p-cluster is given as

$$\frac{\sigma_p}{\sqrt{\sum_j \sigma_{P_j}^2}} \approx C_{P1} \Rightarrow \sigma_p \cong \sigma_{P1} \tag{3.20}$$

Now, revisit the cell's delay sensitivity relation in Equation (3.8). By grouping the sensitivities for switching and non-switching devices in this relation, it can be rewritten as

$$\sigma^2 = \sigma_{N1}^2 + \sigma_{P1}^2 + \sum_{i,i\neq 1} \sigma_{N_i}^2 + \sum_{j,j\neq 1} \sigma_{P_j}^2 \tag{3.21}$$

Using properties ***I*** and ***II*** in the above equation and combining with equations (3.19) and (3.20), it can be re-written as

$$\sigma^2 \approx \sigma_{N1}^2 + \sigma_{P1}^2 \cong \sigma_n^2 + \sigma_p^2 \tag{3.22}$$

Thus, by grouping all nMOS devices into an n-cluster and similarly, all pMOS devices into a p-cluster, the cell's delay sensitivity can be determined by just computing the delay sensitivity of these two clusters.

3.5.1 Handling Multi-fingered devices

Each transistor may have multiple fingers due to several reasons e.g., folding performed during cell layout, handling very wide transistors etc. When there are multiple fingers within a cluster, then all the fingered-devices are connected to same input pin. If this pin is transitioning or switching, then all these fingered devices form switching devices. Hence, all these fingered device fluctuations need to be accounted for. A multi-fingered transistor in the simple approach is handled by treating each finger as a separate device. Since, the fingers have the property that all devices have similar geometry; the cell's delay sensitivity due to each fingered device fluctuation is *almost* same. This property is used to handle multiple fingers. Let f_n (f_p) be number of fingers in the nMOS (pMOS) cluster for the chosen timing arc. Extending equation (3.8) from simple approach for fingered devices, the

cell's delay variance can be rewritten as

$$\sigma^2 = \sum_{f=1}^{f_n} \sigma_{nf}^2 + \sum_{f=1}^{f_p} \sigma_{pf}^2 \approx f_n \cdot \sigma_{n1}^2 + f_p \cdot \sigma_{p1}^2 \tag{3.23}$$

where σ_{nf}, $f = 1 \ldots f_n$ is the delay sensitivity for each fingered device in the n-cluster and, σ_{pf}, $f = 1 \ldots f_p$ is the delay sensitivity for each fingered device in the p-cluster. Using property I and II, the nMOS cluster delay sensitivity in equation (3.16) can be extended for fingered devices as follows,

$$\sigma_n \sim \sum_{f=1}^{f_n} \sigma_{nf} \tag{3.24}$$

Since the impact of each fingered device is equal, σ_{nf} is the same and equal to σ_{n1} for all f_n devices. So, above equation (3.24) can be rewritten as

$$\sigma_n = f_n \cdot \sigma_{n1} \tag{3.25}$$

Similar equations can be derived for the p-cluster with fingered devices. Using equation (3.25) in equation (3.23), the cell's delay variance when the timing arc has multi-fingered devices can be given as

$$\sigma^2 = \frac{\sigma_n^2}{f_n} + \frac{\sigma_p^2}{f_p} \tag{3.26}$$

Thus, for multi-fingered devices, the grouping of a cell into two clusters is still performed. Then, the resulting sensitivity for each cluster is scaled by squared-root of number of fingers corresponding to the timing arc. The advantage of this approach is it reduces the number of characterizations significantly and the run time complexity remains the same as that for single fingered device.

3.5.2 Handling Multiple Direct Channel Connected Components (DCCCs) within Cell

A simple cell in the library is typically a single DCCC, e.g., a NAND/NOR or an inverter (INV) are cells that have devices directly channel connected and form a single DCCC. However, the library may have cells with more than one DCCC within the cell, for example a simple buffer (BUFF) cell with two chained INV and hence has two DCCCs. In the case when a cell has multiple DCCCs, the proposed clustering approach is performed for each DCCC separately. The delay sensitivity for the cell's timing arc is then computed from delay sensitivities obtained for each cluster in each DCCC. That is, if there are q number of DCCCs in a cell, the cell's delay sensitivity can be derived from the following relation

$$\sigma^2 = \sum_q \left\{ \sigma_{nq}^2 + \sigma_{pq}^2 \right\} \tag{3.27}$$

3.5.3 Clustering results in an upper bound

The proposed clustering-based approach results in an upper bound on the delay sensitivity. Due to clustering, all devices within nMOS cluster are varied in same direction and are fully correlated. Using equations (3.16) and (3.17), the cell's equivalent delay variance can be written as

$$\sigma_n^2 + \sigma_p^2 = \left(\sum_i \sigma_{N_i} \right)^2 + \left(\sum_j \sigma_{P_j} \right)^2 \geq \sum_i \sigma_{N_i}^2 + \sum_j \sigma_{P_j}^2 \tag{3.28}$$

Hence the delay sensitivity derived from nMOS and pMOS cluster delay variances is greater or equal to sum of delay variances due to each device fluctuation. Thus, the cluster-based characterization results in an upper-bound on the cell's delay sensitivity. This is typically very useful for timing analysis that require the delay sensitivity estimates to be pessimistic.

Delay sensitivity characterization due to intra-cell mismatch variations can be performed by characterizing for the delay sensitivities of nMOS and pMOS clusters in a combinational cell. Irrespective of the number of devices in the cell, the number of clusters is a constant and is equal to two. Thus, the run time complexity for characterizing n number of local sources of variation is $O(2n)$. The advantage of this proposed cluster-based technique is there is a constant number of simulations required for all cells and is independent of the number of devices in the cell; unlike the methods described in Sections 3.3.1.1 and 3.3.1.2 that depend on number of devices in the cell. Above formulations to compute delay variations for intra-cell mismatch variations were validated using Monte Carlo simulations on simple, multi-DCCC and multi-fingered cells. The results and discussions are presented in the following section.

3.6 Experimental Setup, Results and Discussion

The proposed clustering-based approach has been implemented in an industrial digital library characterization engine. Statistical characterization of each timing arc using cluster-based technique described in previous section was performed for all the cells. The delay sensitivity for each cluster was computed using a finite-difference method. To highlight the effectiveness of the proposed approach we consider here example cells of different types: different device stack configurations, different number of DCCCs, different number of nMOS/pMOS fingers, etc.

For the experiments, the local parameter, ΔL_l corresponding to effective channel length and that for threshold voltage, ΔVt_l were chosen. The parameters were set at $1\sigma =$ 3% of its nominal value. Characterization was carried out by setting one local parameter

at a time, while keeping all other parameters at nominal values and global variations set to zero. Results are illustrated on several SOI and Bulk cells for 65nm technology.

Monte Carlo simulations for each timing arc were performed using the method described in section 3.4, ***Case 1*** for error comparisons. The variance, and hence the standard deviation, was observed by carrying out 3000 iterations within each Monte Carlo simulation. Comparisons of standard deviation from Monte Carlo simulations were performed with standard deviations obtained from proposed clustering-based approach.

Table 3.2: Clustering-based Approach Vs. Monte Carlo: Results for SOI-65nm Cells With Channel Length Mismatch Variations

Cell	Type	Monte-Carlo StdDev (ps)	Clustering StdDev (ps)	Max Error (transition type)	Run time improvement
NAND -2input	1 DCCC f_n=2, f_p=2	0.225	0.233	3.6% (F)	4X
NOR -2input	1 DCCC f_n=1, f_p=1	0.322	0.336	4.0% (R)	2X
BUFFER	2 DCCC f_{n1}=8, f_{p1}=8 f_{n2}=4,f_{p2}=4	0.179	0.184	2.8% (F)	6X
INV	1 DCCC f_n=8, f_p=8	0.106	0.109	2.8% (F)	8X
NAND -Chain	2 DCCC f_{n1}=1, f_{p1}=1 f_{n2}=1, f_{p2}=1	0.318	0.328	3.1% (R)	2X

Table 3.2 illustrates results for SOI library cells at 65nm technology considering channel length mismatch variations while, Table 3.3 illustrates results for Bulk cells at 65nm technology considering threshold voltage mismatch variations. The column "Type" gives details on the type of the cell: the number of DCCCs within the cell is identified and also number of nMOS fingers, f_n and pMOS fingers, f_p is shown. Note that the number of fingers

Table 3.3: Clustering-based Approach Vs. Monte Carlo: Results for Bulk-65nm Cells With Threshold Voltage Mismatch Variations

Cell	Type	Monte-Carlo StdDev (ps)	Clustering StdDev (ps)	Max Error (transition type)	Run time improvement
NAND -2input	1 DCCC f_n=1, f_p=1	5.40	6.04	11.9% (F)	2X
NOR -2input	1 DCCC f_n=1, f_p=1	3.53	3.95	11.9% (R)	2X
BUFFER	2 DCCC f_{n1}=1, f_{p1}=1 f_{n2}=1, f_{p2}=1	3.90	3.97	1.8% (F)	1X
INV	1 DCCC f_n=1, f_p=1	4.76	4.86	2.1% (F)	1X
AND -2input	2 DCCC f_{n1}=1, f_{p1}=1 f_{n2}=1,f_{p2}=1	2.88	3.05	5.9% (R)	2X
AND -3input	2 DCCC f_{n1}=2, f_{p1}=2 f_{n2}=4,f_{p2}=4	1.40	1.51	8.0% (R)	7X
AOI	1 DCCC f_n=4, f_p=4	1.20	1.33	10.8% (F)	8X

for transistors connected to each input pin may vary; however, only fingers corresponding to the input pin of the chosen timing arc is given. The next column "Max Error (transition type)" shows percentage error in the proposed approach and the output transition that contributes to max error. The error is computed with Monte Carlo simulation as baseline to understand the accuracy of the proposed approach. The transition type can be either falling (F) or rising (R). The last column provides the run time improvement factor when comparing with the simple approach as described in section 3.3.1.1. The run time improvement when comparing with Monte Carlo approach is $\approx 1500X$ for all cells and is not shown in the tables.

Consider for example, the results in Table 3.2 for the 2-input NAND cell. This cell

is a single DCCC and has two nMOS and two pMOS fingers. The maximum error for this cell is 3.6% and it corresponds to the timing arc that results in the output to be a falling (F) transition. It can be observed that the maximum error for all the SOI cells is within 4%, and that for all Bulk cells is within 12%. The run time improvement for the proposed technique is as much as $8X$. It can be observed that the run time improvement is very high for multi-DCCC and multi-fingered cells. The clustering-based approach thus achieves a very high run time advantage with acceptable level of loss of accuracy.

It can be observed that the error is largest for the threshold voltage variations, specifically for transition type that is controlled by series devices (e.g., falling transition for NAND). The reason is that the threshold voltage variation on each device is dependent on the effective resistance of source and drain regions in addition to the channel dimensions. For series devices this causes the variations in threshold voltage to become correlated.

3.7 Correlations Due to Bias Conditions

For transitions with series devices in the conduction path, there is an increase in error. This error is due to the fact that the cluster model does not capture the staggered change in the states of stacked transistors. For example, if M_{n_1}, M_{n_2} are the stacked transistors of a 2-input Nand cell. When input to M_{n_1} rises, transistor M_{n_1} turns on and starts to discharge node N_1 between M_{n_1} and M_{n_2}. However, the transistor M_{n_2} does not conduct until the internal node N_1 discharges to $V_{dd} - V_{th}$. If M_{n_1} is perturbed slightly, then the loading seen on M_{n_2} is going to be different. Consider two cases where the top transistor, M_{n_2} is switching and perturbed by $\Delta V_{th} = 3$:

- Case(a): the bottom transistor is at nominal bias, that is, $\Delta R_{bottom} = 0$.

- Case(b): the bottom transistor is set to a perturbed state, that is, $\Delta R_{bottom} = 3$.

Both these cases are illustrated in Figure 3.14. Ideally, for random variations, both case (a) and (b) should have same transfer characteristics. However, for V_{th} variations, there is an inter-dependence on the bias (or state) of the transistors. What does this mean? If the top transistor is perturbed by the settings described in Case(a) and Case(b) then the I-V characteristics of the devices change significantly (illustrated in Figure 3.15). The drain-to-source resistance, R_{ds} of the bottom device changes from Case(a) to Case(b) resulting in delay variations due to top transistor V_{th} variations correlated with the delay variations due to bottom transistor variations.

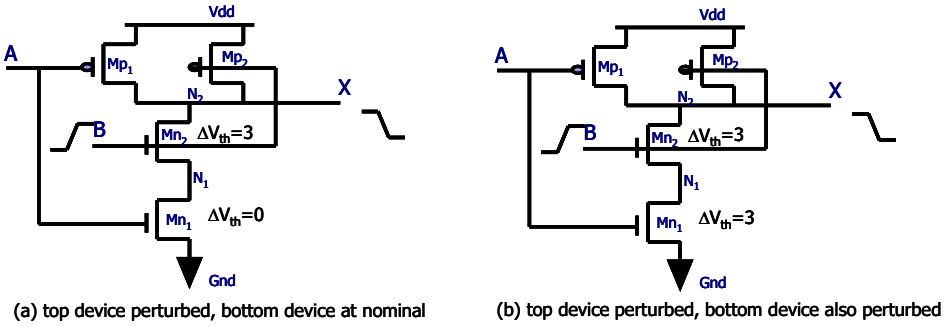


Figure 3.14: Illustration of Bias Conditions in Stacked/Series Transistors

Consider however, the case where a perturbation on the L_{eff} parameter is performed. Again by switching the top transistor and setting bottom transistor to case (a) and case (b). Then, the I-V characteristics of the devices do not change significantly. This can

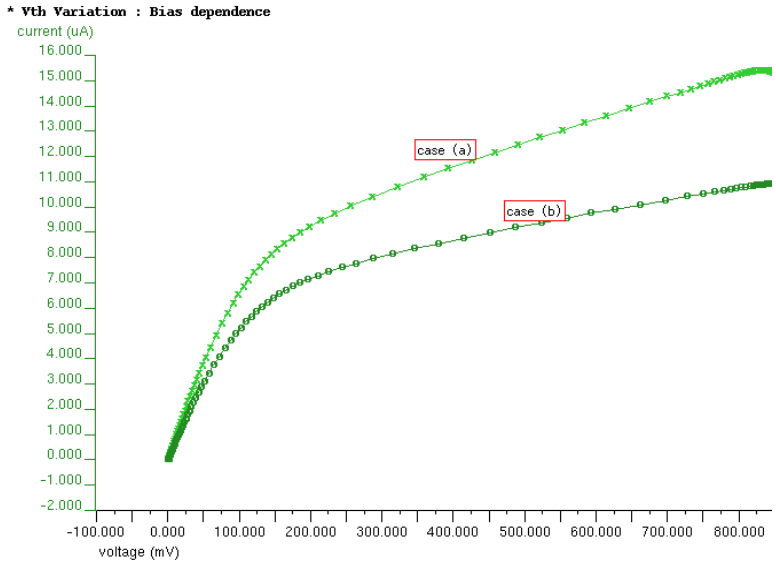


Figure 3.15: Stacking Effect for V_{th} Variations: I_{ds} vs. V_{ds} for Stacked Transistors With Bias Conditions in Case(a) and Case(b)

be clearly observed in Figure 3.16. Similar observations can be made for other within-die random variation parameters. What is happening for these parameters? The drain-to-source resistance, R_{ds} of the devices does not depend on the states (or bias conditions) of the other series-devices. That is, there is no significant change in R_{ds} between Case(a) and Case(b).

This indicates that V_{th} variations are a special case, where the delay variations due to V_{th} variations on one device has dependence (are correlated) on the bias conditions of other devices in the stack. For parameters that have such dependence on the bias of other devices a characterization of the correlation need to be performed ahead. Since, the correlation dependence on the input slew and output loading conditions is of a higher order, a one time correlation characterization for the cluster that forms stacked devices can be performed ahead of time.

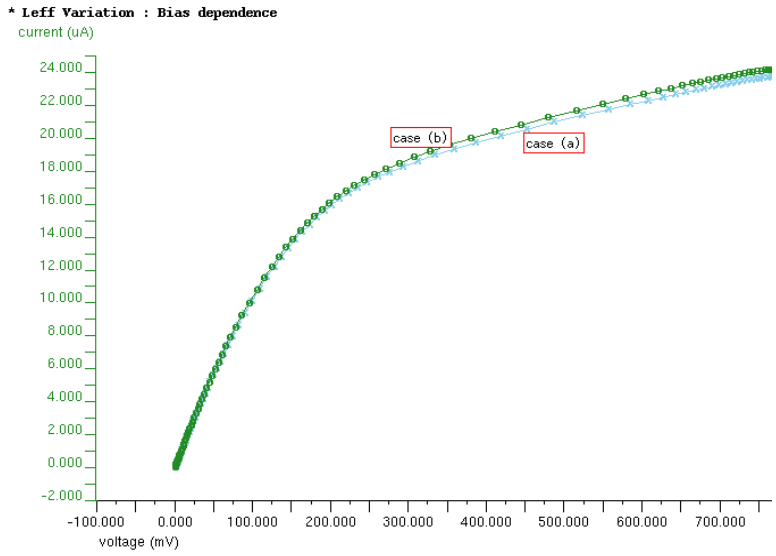


Figure 3.16: Stacking Effect for L_{eff}: I_{ds} vs. V_{ds} for Stacked Transistors With Bias Conditions in Case(a) and Case(b)

3.8 Conclusions

In this chapter, a clustering based approach to account for intra-cell mismatch variations and characterize for delay variations is discussed. The clustering-based approach easily extends for handling multi-fingered devices by keeping the run time the same as that for single fingered device. The result is the clustering technique achieves significant run time improvement for all combinational cells in the library. Further, it can be observed that the clustering-based approach always has constant a number of clusters for each cell and hence, a constant run time independent of number of devices in the cell.

The proposed clustering approach cannot be applied directly to clocked logics including sequential logic, dynamic and domino logic etc. Each logic family has to be studied for both the types of the logic as well as the types of performance metrics that need to

be handled during timing. A method to perform statistical characterization of timing constraints for sequential cells, specifically flip-flops is described in the following chapter.

Chapter 4

Statistical Constraint Characterization

4.1 Overview

Gate-level SSTA requires all standard cells, both combinational and sequential cells to be characterized for delay sensitivities. Additionally, the sequential cells need to be characterized for constraint sensitivities to several process parameter variations. The variations in constraints due to both global (inter-die) and mismatch (local-random) variations need to be included during statistical timing analysis. A straightforward approach to computing flip-flop constraint variability is to perform finite-difference on constraints obtained using search-based approach. In this approach, a perturbed parameter value is assigned to the flip-flop and the corresponding constraint (either setup time or hold time) is computed for the parameter variation. For each perturbed parameter value, a search based technique is used to characterize for the constraints. Typically, each such search requires $16 - 20$ simulations. With the increase in the number of parameter variations, the runtime complexity of constraint sensitivity characterization can become very expensive. If there are p parameters, computing the constraint sensitivity requires p characterizations in addition to the nominal constraint characterization; that means, this requires additional $20p$ simulations (each characterization requires as much as 20 simulations). Further, these additional characterizations need to be repeated for different data and clock slew combinations. For mismatch/local variations, each device in the flip-flop needs to be treated as a separate

parameter of variation [59] [60] [61]. The problem of constraint sensitivity characterization becomes extremely expensive when considering mismatch parameters.

In this chapter we present a delay-based approach to characterize for constraint sensitivities to global and mismatch variations. The basic idea in this approach is to use the fact that setup and hold times of a sequential cell are a function of the internal data and clock propagation delays. Using the variations in the propagated delay, variations in setup and hold times can be computed very accurately. We show that the proposed approach to computing constraint sensitivities is more robust than the search based methods. We also present modified method to achieve runtime optimizations for characterization of mismatch constraint sensitivities.

Section 4.2 describes the preliminaries for constraint characterizations. In this section, the current statistical characterization approach and challenges in constraint sensitivity characterization are described. Section 4.3 presents the proposed constraint sensitivity characterization approach and the experimental setup and results for 45nm technology cells are discussed in Section 4.4. Recommendations for future work are discussed in the last section.

4.2 Background and Prior Work

4.2.1 Terminology

Latches and flip-flops (edge-triggered latches) are the sequential circuit elements used in synchronous designs. These elements are characterized using the following timing metrics.

- **Setup Time,** T_s, is the amount of time that the data must be stable before the capturing clock edge.
- **Hold Time,** T_h, is the amount of time the data must remain stable after the capturing

clock edge.

- **Clock2q Delay,** T_{c2q}, is the propagation delay from the capturing clock edge to a valid output transition. The clock2q delay for a large setup time is termed as stable clock2q delay, T^0_{c2q}.

The quantities T_s, T_h and T_{c2q} are interdependent. As T_s decreases, there is a delay push out resulting in increase in T_{c2q} delay. For a predefined T_{c2q} delay, the hold time increases with a decrease in T_s. There is a minimum setup time independent of hold time and is represented as T_{sm}. Similarly, minimum hold time independent of setup time is represented as T_{hm}. We use master-slave flip-flops to illustrate the formulations throughout the rest of the document (unless otherwise specified).

4.2.2 Nominal Constraint Characterization

Gate-level static timing analysis (STA) requires that the sequential cells are characterized for two constraints: setup time and hold time. Typically these constraints are pre-characterized and stored in a two-dimensional table indexed by data slew and clock slew. Each sequential cell is characterized using a transistor level circuit simulator (e.g., spice simulator). This characterization procedure is repeated for each data transition edge, either a rising edge or a falling edge.

The basic idea behind setup and hold time characterization for STA is to capture the stable operating region of a flip-flop or latch. During timing analysis the constraints ensure that the flop does not fall into its failure region. In the stable region, the clock2q delay is T^0_{c2q}. If the data to clock skew is very small then the flop fails to latch the data or fails to propagate the correct data – this region is termed as failure region. Figure 4.1

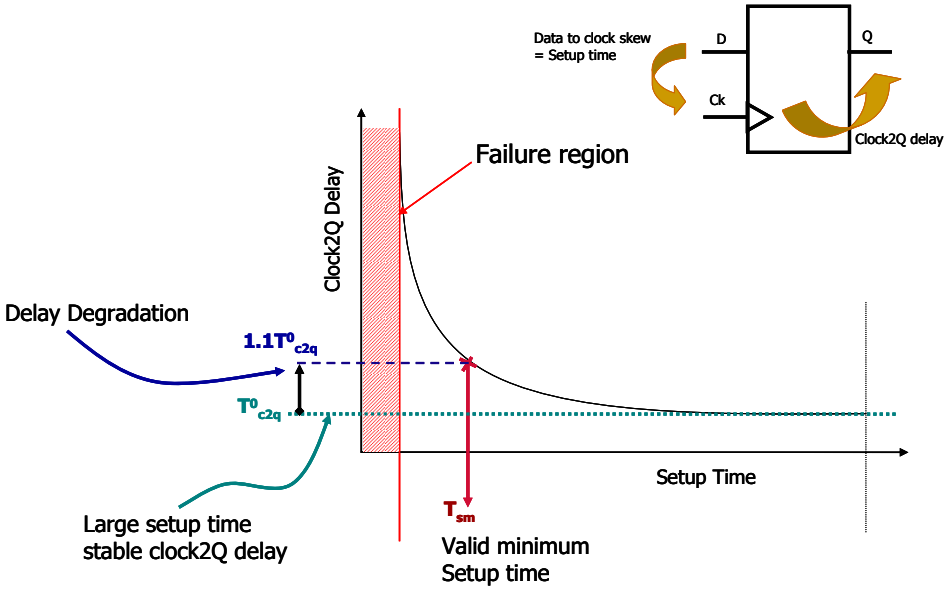


Figure 4.1: Flip-Flop Basics: *Delay-degradation*, Minimum Setup Time and Failure Region

illustrates different regions of flip-flop. The region between stable and failure region (where the master latch does not have a stable data signal) is termed as the metastable region. Typically, setup (hold) time characterization is performed such that the data to clock skew crosses over from the metastable into the stable region. There are different approaches to identify this crossover point. One approach is to identify the point where $T_s + T_h$ is minimized. Another approach, commonly used for high-performance circuits, proposed in [62] is to determine the time where $T_s + T_{c2q}$ is minimized. This would be a point where slope of T_s vs. T_{c2q} becomes -1.

A common approach used in industry is to determine the setup (hold) time with a

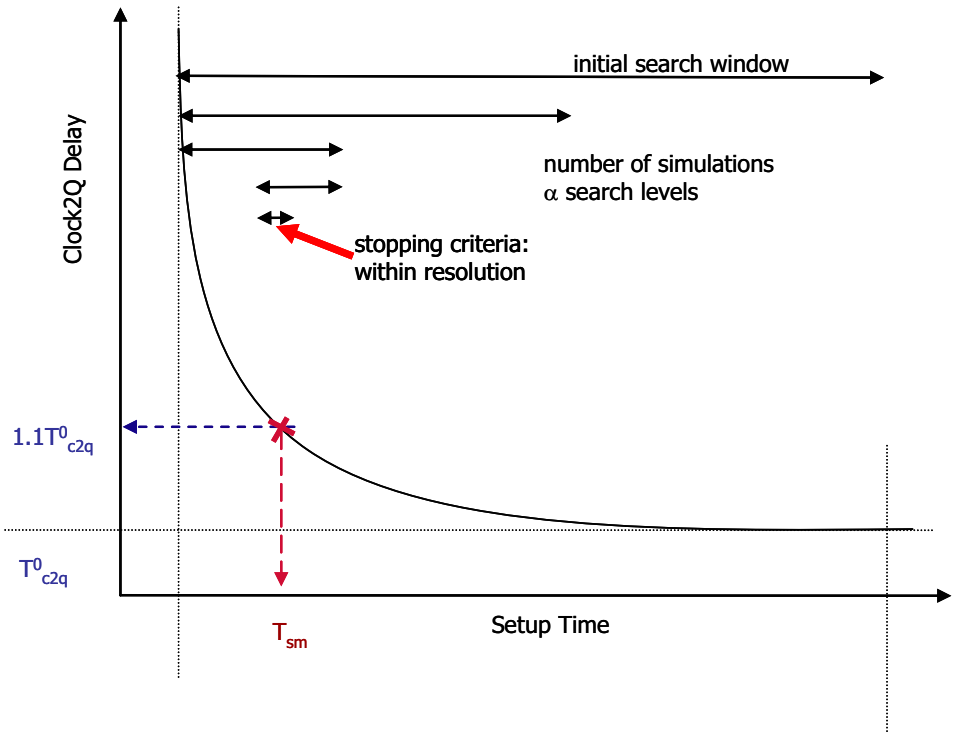


Figure 4.2: Illustration of Search-based Method (Using Binary Search)

5-10% degradation in the T^0_{c2q} delay. We term this approach as *delay-degradation* method and use this for all comparisons. The nominal value for setup and hold time constraints is then characterized using either a binary search or guided search method. In a binary search method, an initial search window between the stable and failure region is estimated and then a binary search within this window is performed to determine the crossover point. We term this method of estimating the minimum setup time as *sbSetup*. The value estimated using this method is T_{sm}. These are illustrated in Figure 4.2.

4.2.2.1 Search-based Method

To characterize for setup time using a search based method, the data to clock skew is varied till the flop no longer captures a valid output or the clock-to-output degrades by a predefined degradation value (e.g., 10% of T^0_{c2q}). A simple search is to sweep the skew from a large value at a predefined resolution. The search may be refined using binary search or some guided search ([63], [64]) to improve run time. On an average the search based techniques require $16-20$ transient simulations to determine the actual setup time. Further, in a search based technique a predefined resolution (termed as *length* in [63]) is required for the search. In guided search techniques ([63], [64]), the basic idea is to pre-determine the interdependent setup-hold curve for a given T_{c2q} degradation and then, compute the setup-hold pairs by tracing this curve. For example, in [64] a Backward Euler based approach is used to trace the setup/hold curve. Such dependent setup and hold time pairs are then used during gate level static timing analysis.

4.2.3 Constraint Sensitivity Characterization

Since the constraints (setup time or hold time) are the result of the difference in data and clock propagation delays, the constraint sensitivities to variation parameters are generally small. While the sensitivities to each parameter may be small, it is still of interest to characterize these sensitivities. This is because with each technology generation, there are large numbers of parameters of variation added to the spice models and the constraint sensitivities accumulate with such increase in number of variations.

As discussed in the previous chapters, process disturbances are often described by device parameter variations which can be classified into two types, (a) *Global variations* are inter-

die variations, that are same for all devices on a chip, and (b) *Mismatch variations* are intra-die [1] random variations, that vary from device to device within a chip.

Let $\Delta\overline{X} = \{\Delta X_1, \cdots \Delta X_m\}$ be m global variations. Let $\Delta\overline{R} = \{\Delta R_{11}, \cdots \Delta R_{np}\}$ be $n \times p$ mismatch variations where, p is the number of devices in a cell and n is the number of intra-die variation parameters. Both global and mismatch variations are modeled as a standard normal distribution, $N(0,1)$ and these are statistically independent to each other. Since the statistical variations are often much smaller than the nominal parameter values, we use a first-order formulation for the constraint sensitivities, represented as follows,

$$\Gamma = \Gamma_0 + \sum_{i=1}^{m} \Gamma_i \Delta X_i + \sum_{j=1}^{n} \sum_{k=1}^{p} \gamma_{jk} \Delta R_{jk} \tag{4.1}$$

where Γ_0 is the nominal constraint and is characterized by setting all variations to zero. The quantities, Γ_i and γ_{jk} are constraint sensitivities with respect to ΔX_i and ΔR_{jk} respectively. These quantities are obtained as a result of statistical characterization of constraints. It is important to understand that from a cell characterization perspective, each variable in $\Delta\overline{X}$ impacts all devices identically and hence, represents a single cell-level variable. However, each local variable in$\Delta\overline{R}$ represents a separate random variable for each device in the cell. Thus, characterization of constraint sensitivities due to global variations (global sensitivities) is performed by varying a given parameter for all devices in a cell; while, that due to mismatch variations (mismatch sensitivities) is performed by varying a given physical parameter assigned to one device at a time in the cell. For the purpose of statistical timing analysis, the variables, $\Delta R_{jk}, \forall k$ can be combined to represent as a single random variable, ΔR_j with equivalent sensitivity, $\gamma_j = \sqrt{\sum_{k=1}^{p} \gamma_{jk}^2}$. This is possible because these variables

[1] intra-die variations may also have a spatial component which is not described here for simplicity

are statistically independent.

A simple and straightforward approach to obtaining Γ_i, γ_{jk} is use of a finite-difference approach[2] to compute the constraint sensitivities. In this approach, each parameter, ΔX is perturbed one at a time and setup time computation is repeated for the new circuit conditions. For each perturbation, the search based algorithm described in previous section 4.2.2 is applied to compute the setup time. The difference in the perturbed setup time, $T_{sm}^{'}$ and the nominal value, T_{sm} determines the deviation in the setup time due to ΔX. While this is a straightforward approach, there are several challenges in using search based algorithms for constraint sensitivity computation.

- It requires several tens of transient simulations to characterize for constraint sensitivity to each parameter. This can become infeasible when considering a large number of variation parameters or when the number of devices in the cell is large (for mismatch variations).

- In a search-based technique, the search stops when the data to clock skew reaches a certain pre-defined search resolution. If the sensitivities are smaller than the search resolution, then this approach will result in either underestimating or overestimating the sensitivities by an amount equal to the difference between the actual value and the resolution. The resolution may be set to a very small value, but this will increase the number of search iterations. Figure 4.3 illustrates setup time sensitivities computed with a $2ps$ resolution for a $65nm$ technology flip-flop. The figure demonstrates that using a search-based method the setup time sensitivities snap to the resolution grid

[2]Monte Carlo will be used as the baseline method for accuracy comparisons

of $2ps$. This indicates that in order to obtain accurate and reliable sensitivities, a higher resolution may need to be implemented, which may further increase the number of simulations. So a method that does not depend on the resolution is desired to characterize for constraint sensitivities.

- Parameter variation results in both setup time and clock2q delay change. During statistical timing it is desirable to either have constant T_{c2q} or constant slope between setup time and clock2q delay across the parameter range. Since clock2q changes with parameter, the *delay-degradation* with parameter changes may not represent a constant slope. If, however, T_{c2q0} from nominal conditions is used to determine the crossover point at perturbed parameter value, then the flop may move into its failure region. This means, for the perturbed parameter value the setup time variations cannot be determined correctly using *sbSetup* method.

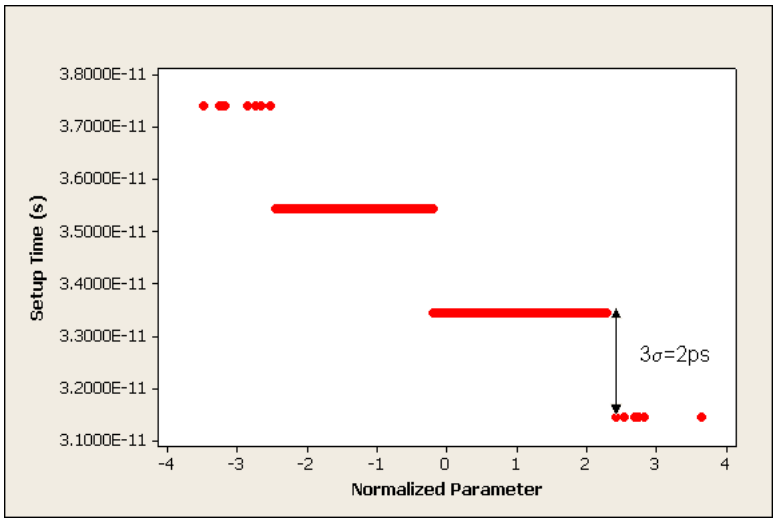


Figure 4.3: Setup Time Using *sbSetup* Method With $2ps$ Resolution

The proposed approach addresses these problems by computing propagated delays that have a continuous search space. The approach is independent of any resolution factors with continuous values for sensitivities. Thus, it results in more robust sensitivity values. Further, the number of simulations required for each parameter of variation is at the most two additional simulations – reducing the overall runtime for constraint sensitivity characterization significantly. The approach is described in the following section.

4.3 Proposed Approach

4.3.1 Delay-based Constraints

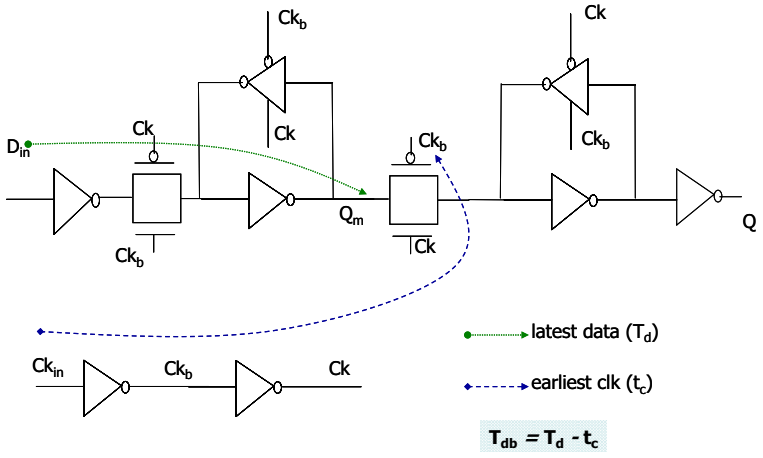


Figure 4.4: Data and Clock Paths for *dbSetup*

Consider a typical master-slave flip-flop (MSFF) configuration illustrated in Figure 4.4. The MSFF has master latch output, Q_m and slave latch output (also flip-flop output) Q. Let T_d represent the latest data propagation delay from D_{in} to Q_m. By definition, the setup time is the minimum amount of time for the data to be stable such that, when the clock

transitions, this stable data can be propagated further to the output of the flip-flop, Q. Let the clock edge occur at zero time. Then, T_d determines the time required for the data to be stable and available before the clock reference edge occurs and is the minimum setup time.

In reality, the clock signal has a certain propagation delay, t_c from Ck_{in} to Ck_b. Consequently, the minimum setup time shifts by this amount (illustrated in Figure 4.5). We term this method of finding the minimum setup time as *dbSetup*. Thus, a minimum setup time T_{db} can be determined using the difference in the latest data propagation delay, T_d and the earliest clock propagation delay, t_c and is given as: $T_{db} = T_d - t_c$.

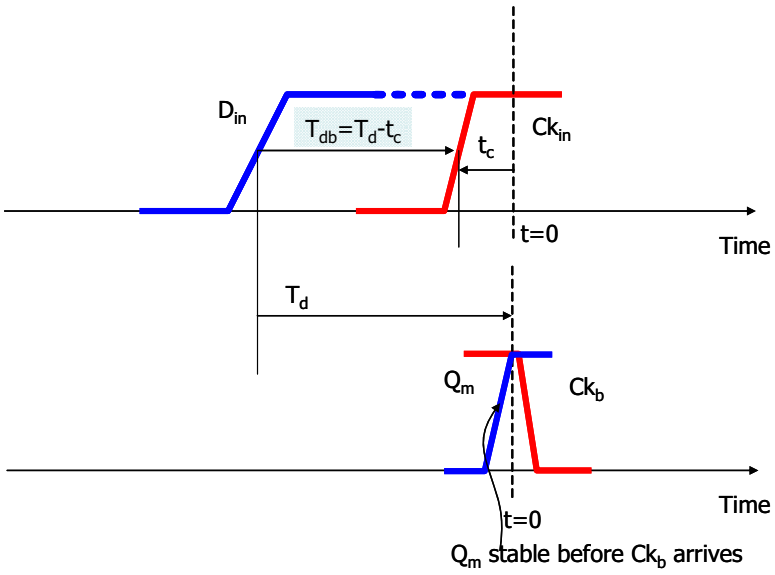


Figure 4.5: Illustration of Min Setup Time

A similar formulation can be used for computing the minimum hold time using the latest clock edge, T_c and an early data propagation delay, t_d. The data and clock paths for hold time computation may be different from those for the setup time computation. The paths are illustrated in Figure 4.6 and the waveforms to compute minimum hold time are

depicted in Figure 4.7. The minimum hold time is then given as $H_{db} = T_c - t_d$.

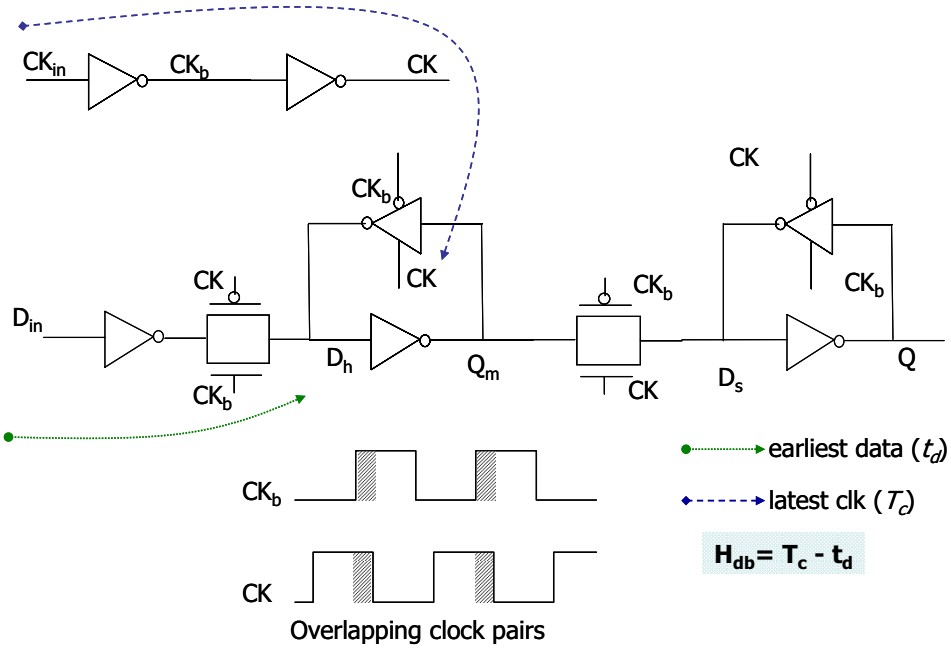


Figure 4.6: Data and Clock Paths for *dbHold*

Due to this difference in delays, both setup time and hold time can be either positive or negative values. Using the above formulations, characterization of setup/hold time requires at the most two simulations (there are ways where this could be achieved in single simulation as well and is not discussed here) to determine the data and clock propagation delays. It is however difficult to identify the right combination of switching thresholds for data and clock signals that result in T_{db} to be equal to T_{sm}. The switching thresholds depend on the feedback structure in the master latch and also depends on the arrival time and slew of the clock and data signals. Consequently, only search-based techniques are generally used to determine, T_{sm}. We empirically show that *dbSetup* method can be used more effectively for

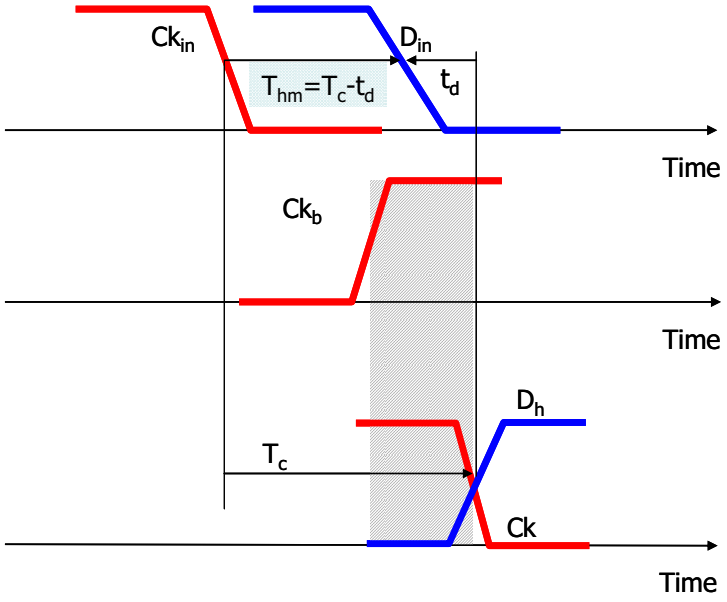


Figure 4.7: Illustration of Min Hold Time

constraint sensitivity characterizations.

Correlation between *dbSetup* and *sbSetup* Inherently, the two methods *sbSetup* and *dbSetup* are different. The *sbSetup* method probes data transition at the output, while the *dbSetup* method uses a stable data signal at the internal nodes of latch/flip-flop to determine the minimum setup time. In order to understand the correlation between setup time measurements from *sbSetup*, T_{sm}, and *dbSetup*, T_{db}, we performed several Monte-Carlo simulations using global and mismatch parameter variations. Within each Monte-Carlo iteration, we compute both *sbSetup*, T_{sm} and *dbSetup*, T_{db} for given clock and data slews. The T_{sm} is computed for a large clock2q *delay-degradation* - the reason for this is for large delay-degradation, the setup time is independent of clock2q delay. Dependence of setup time on clock2q delay is described in detail in Section 4.3.3. To get very high accuracy on

the T_{sm} values, we used a search resolution of 0.01ps. Figures 4.8 and 4.9 illustrate the results for the case where gate-length variations (as global parameter) with the 3σ value set at 10% of its nominal value was used. It can be observed that there is a very high correlation between T_{sm} and T_{db}. Further, the variation of T_{db} with respect to each parameter variation (in Figure 4.9) is approximately same as the variation of T_{sm}.

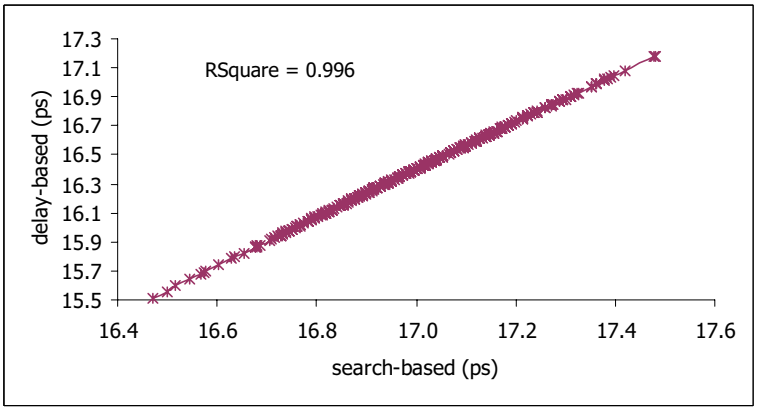


Figure 4.8: T_{sm} vs. T_{db}: Monte-Carlo results

Thus, T_{sm} is highly correlated to T_{db} and T_{db} can be used to estimate T_{sm}. Further, the sensitivities using delay-based method are very close to sensitivities obtained using search-based methods across the whole range of parameter variations.

4.3.2 Sensitivity Using Delay-Based Approach

The basic idea in the proposed approach is to equate the constraint sensitivity to be the same as the sensitivity obtained due to the delay-based method. More formally, for a given parameter variation, ΔX, this implies,

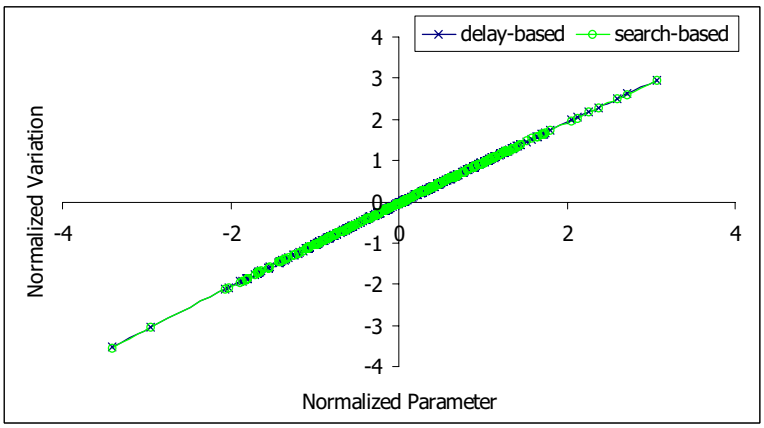


Figure 4.9: Sensitivity of T_{sm} and T_{db}

$$\Delta T_{sm}|_{\Delta X} \approx \Delta T_{db}|_{\Delta X} \tag{4.2}$$

Using first-order formulation, the delay-based sensitivity can be represented as follows,

$$\Delta T_{db} = \sum_{i=1}^{m} \xi_i \Delta X_i + \sum_{j=1}^{n} \sum_{k=1}^{p} \rho_{jk} \Delta R_{jk} \tag{4.3}$$

where ξ_i and ρ_{jk} are sensitivities with respect to global parameters ΔX_i and mismatch parameters, ΔR_{jk}, respectively. As discussed in the previous section, the delay-based setup time is the difference in propagated delay in data signal and propagated delay in clock signal from external node to the internal node. Thus, the delay-based sensitivities ξ_i and ρ_{jk}, can be derived using the data and clock path sensitivities and are described below.

Global Sensitivities: From a cell characterization perspective, global variations, ΔX_i impact all devices. Hence, a single random variable for each ΔX_i is assigned to all devices. By computing the data path sensitivities, d_i and clock path sensitivities, c_i with respect to

ΔX_i, the constraint sensitivities can be obtained as

$$\xi_i = d_i - c_i \tag{4.4}$$

Mismatch Sensitivities: For mismatch variations, the sensitivities are determined by setting one random variable, ΔR_{jk} to each device $'k'$. Again using the data and clock path sensitivities, d_{jk} and c_{jk}, respectively, the mismatch sensitivity with respect to ΔR_{jk} can be determined using the difference in these as $\rho_{jk} = d_{jk} - c_{jk}$. Since ΔR_{jk} are statistically independent, for a given parameter, $'j'$, an equivalent sensitivity considering all device variables together can derived as

$$\rho_j = \sqrt{\sum_{k=1}^{p} \rho_{jk}^2} = \sqrt{\sum_{k=1}^{p} \{d_{jk} - c_{jk}\}^2} \tag{4.5}$$

Using the equivalent sensitivities for mismatch variations, the equation in 4.3 can be rewritten as:

$$\Delta T_{db} = \sum_{i=1}^{m} \xi_i \Delta X_i + \sum_{j=1}^{n} \rho_j \Delta R_j \tag{4.6}$$

where ΔR_j is the equivalent random variable. From the above equations it can be observed that, the global sensitivities of data and clock path cancel each other; while the equivalent sensitivities statistically add up for the mismatch parameters. This results in setup time sensitivity to be smaller for global variations compared to that for mismatch parameters. Similar formulations can be derived for hold time global and mismatch sensitivities.

Thus, the proposed approach requires just two additional simulations for each parameter of variation, one each to compute the data propagation delay and clock propagation delay. For search-based methods, each parameter variation may require $16 - 20$ additional simulations. Using delay-based method to compute the constraint sensitivities, there is a

runtime advantage of $8X - 10X$ for each parameter of variation. Typically, the data and clock paths are independent and hence, the data propagation delay does not depend on the clock slew and the clock propagation delay does not depend on the data slew. Figure 4.10 illustrates the comparison of run time improvement that is possible using different search-based methods versus the proposed delay-based method for global sensitivities. Further, since the sensitivities are computed using the delays, there is no requirement for setting any resolution factors for the proposed approach and, hence the sensitivities are more robust.

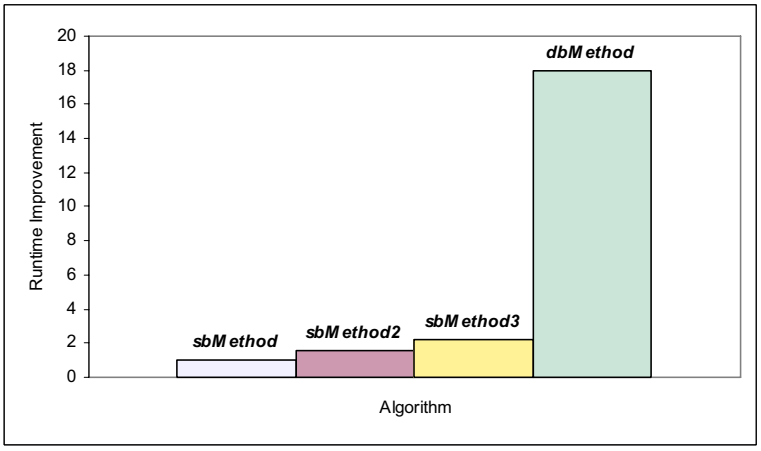


Figure 4.10: Comparison of Run time for Different Constraint Sensitivity Characterization Algorithms

4.3.3 Degradation Contour

Setup time and clock2q delay are dependent quantities. These are also a function of the parameters of variation. That is, T_{sm} can be represented as a function, $f(\Delta\overline{X}, \Delta\overline{R}, T_{c2q})$. With a decrease in setup time, the data and clock signals get closer resulting in an increased

clock2q delay. As discussed earlier in Section 4.2, the minimum setup time T_{sm} is computed for a given degradation in clock2q delay.

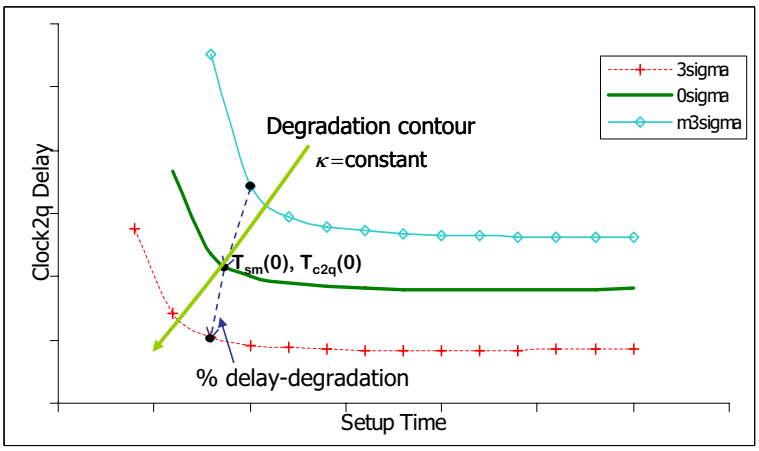


Figure 4.11: Variations in T_s Vs. T_{c2q}

Now, consider a random parameter, ΔX with standard normal distribution $N(0,1)$. Applying a certain perturbation to this parameter results in a variation in both setup time and clock2q delay. Figure 4.11 illustrates clock2q delay vs. setup time for three discrete values of ΔX: -3, 0 and 3. The minimum setup times computed using 5% *delay-degradation* are given by points $T_{sm}(-3)$, $T_{sm}(0)$ and $T_{sm}(3)$, respectively. The corresponding clock2q delays are $T_{c2q}(-3)$, $T_{c2q}(0)$ and $T_{c2q}(3)$. It can be observed that the whole clock2q delay vs. setup time curve shifts with parameter change. Therefore, in order to find the T_{sm} for a particular parameter, the clock2q for that parameter value need to be pre-determined. If the setup time variation is computed by using the nominal clock2q delay, $T_{c2q}(0)$, for different parameter values, then the flop may move into its failure region. Further, the slope between

setup time and clock2q delay is not the same for these three points, which is not desirable when computing the sensitivities. Figure 4.12 illustrates the slope between setup time and clock2q delay as a function of setup time. For setup time less than a certain minimum value, the slope is zero and below that value the flop enters the failure region. As the setup time increases, the slope is negative and decreases further.

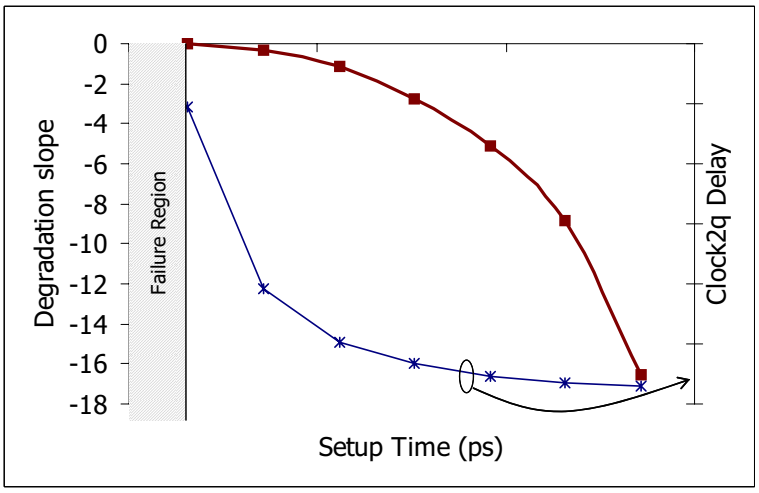


Figure 4.12: Setup Vs. Clock2Q Slope

To capture the clock2q dependence accurately, we define a constant slope contour for the setup vs. clock2q delay curve across parameter variations. We term the constant slope between setup and clock2q as *degradation slope* and define it as $\kappa = \frac{\partial T_s}{\partial T_{c2q}}$. The constant slope contour across parameter variations is termed as *degradation contour*. By choosing a constant setup time vs. clock2q delay slope, we ensure the crossover point across parameter variations and the flop does not go into its failure region. The *degradation slope*, κ may be computed at the delay-degradation point during nominal setup time characterization. This

allows the nominal setup time characterization to be unchanged. The sensitivities are then computed using this slope for all parameter variations.

Let q_i, ϖ_{jk} be clock2q delay sensitivities to global and mismatch variations respectively. Let ΔT_{c2q} be represented as first order formulation similar to ΔT_{db}

$$\Delta T_{c2q} = \sum_{i=1}^{m} q_i \Delta X_i + \sum_{j=1}^{n} \sum_{k=1}^{p} \varpi_{jk} \Delta R_{jk} \tag{4.7}$$

Setup time variations considering *degradation-contour* is then given as

$$\Delta T_{sm} = \Delta T_{sm}|_{\Delta X} + \kappa \cdot \Delta T_{c2q} \tag{4.8}$$

When the *delay-degradation* is large, then $\kappa \approx 0$ and above equation becomes: $\Delta T_{sm} \approx \Delta T_{sm}|_{\Delta X}$. However, from the previous Section 4.3.2, for large *delay-degradation* $T_{sm}|_{\Delta X}$ was equated to the delay-based sensitivity, ΔT_{db}. Thus, the setup time variation equation can be rewritten as

$$\Delta T_{sm} = \Delta T_{db} + \kappa \cdot \Delta T_{c2q} \tag{4.9}$$

The above equation includes sensitivities to both global and mismatch variations. The *degradation-slope* is computed once during the nominal characterization and requires no additional simulations.

4.3.4 Runtime Optimization for Mismatch Sensitivities

For mismatch variations, the constraint sensitivities need to be computed by assigning random variables to each device in the cell. When considering latches/flip-flops, the number of devices and hence, the number of mismatch variables can be large (40-100 devices) making

the problem of computing constraint sensitivities infeasible. Even though the constraint sensitivity is represented finally using a single variable, ΔR_j, computing the sensitivity to each physical parameter requires an additional *2p* simulations (where, p is the number of devices in the flop). In the worst-case, the runtime complexity of characterizations becomes *O(2np)*, where n is the number of intra-die physical parameters. To address this problem, we first analyze the sensitivity contributions from each device due to each parameter. Equation 2.32 can be rewritten as:

$$1 = \frac{\rho_{jmax}^2}{\rho_j^2} \sum_k \frac{\rho_{jk}^2}{\rho_{jmax}^2} \Rightarrow 1 = C \cdot \sum_k \{\frac{\rho_{jk}}{\rho_{jmax}}\}^2 \tag{4.10}$$

where $\rho_{jmax} = max_{\forall k} \rho_{jk}$ and C is a constant. Thus, ordering of devices in terms of the ratio $\frac{\rho_{jk}}{\rho_{jmax}}$ provides a ranking of devices in terms of their contribution to the equivalent sensitivity, ρ_j. Consequently, we define two metrics.

Relative sensitivity: (S_{jk}) is the ratio of constraint sensitivity obtained due to perturbation from each device, k and the maximum constraint sensitivity, given as: $S_{jk} = \frac{\rho_{jk}}{\rho_{jmax}}$.

Most sensitive devices(MSD): These are devices that have relative sensitivity, $S_{jk} > 0.1$. MSD and S_{jk} together provide metrics to determine the significance of each device contribution to the total sensitivity ρ_j.

Figure 4.13 illustrates S_{jk} for each device in an MSFF configuration. The following observations can be made from the relative sensitivity analysis.

- Fewer than 15% devices have relative sensitivities greater than 0.1.
- Contribution of MSD is larger than 97% to the total sensitivity.

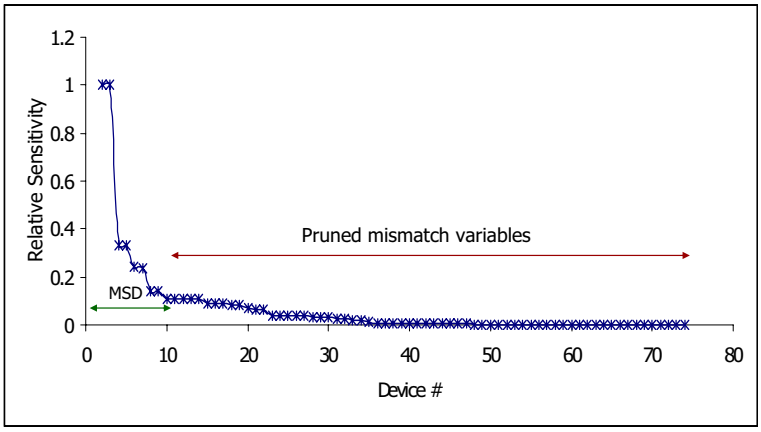


Figure 4.13: Relative Sensitivity for Each Device

- For master-slave configurations, only devices in the master latch form the set of MSD.

Using above observations, we select only the MSD and prune all the remaining devices during computation of mismatch sensitivities. This results in significant runtime improvements and can be estimated to be $\approx 7X$ (using 100%/15%) when compared without pruning of variables. Thus, the total runtime improvement for computing constraint sensitivities to mismatch variations using proposed method at the minimum is $\approx 70X$ when compared to *sbSetup*. Note that the number of parameters required to compute the mismatch sensitivities for data and clock paths are generally mutually exclusive. This property may be used for further runtime optimizations. We term this method with pruning of MSD enabled as *dbSetup-opt* method.

4.4 Results and Discussion

The proposed constraint sensitivity characterization was implemented within an industrial statistical characterization framework. For mismatch sensitivities, the proposed approach with pruning of mismatch variables as discussed in Section 4.3.4 was also implemented. One of the main challenges in implementing the *dbBased* method is to identify the data and clock paths accurately. These paths are different for setup and hold time computations. We identified these paths in advance and annotated them as inputs to the characterization engine. Both Monte-Carlo simulations and search-based constraint sensitivity characterization were implemented for comparisons. To illustrate the results, each flop cell chosen from a $45nm$ SOI technology library was representative of a separate flop family. Further, we chose a single global parameter and a single intra-die/mismatch variation parameter: namely, (a) gate-length variation and (b) threshold voltage variation, respectively. The 3σ of these variations are set at 10% its nominal values.

Table 4.1: Setup Time Sensitivities for Global Variations

Cell	MC Dev(ps) *I*	*sbMethod* Dev(ps)/Error *II*	*dbMethod* Dev(ps)/Error *III*	Runtime *sbSetup* *IV*	Runtime *dbSetup* *V*
FF1	0.58	0.56 / 5%	0.61 / 5%	15.3	1.77 (9X)
FF2	0.68	0.68 / 0%	0.57 / 16%	30.13	2.76 (11X)
FF3	0.8	0.84 / 5%	0.8 / 0%	24.07	1.8 (13X)
FF4	0.58	0.65 / 12%	0.55 / 5%	19.8	2.38 (8X)

Baseline sensitivities were obtained using exhaustive Monte-Carlo simulations for both global and mismatch sensitivities. For search based methods, the resolution was chosen to be 0.01ps. Table 4.1 shows results for setup time sensitivities from global variations and Table 4.2 gives results for mismatch variations. In these tables, columns I, II, III show the sensitivities for Monte-Carlo (MC) simulations, traditional search-based (*sbSetup*)

method, proposed delay-based (*dbSetup*) method respectively. The run time for computing sensitivities to global variations in cpu-units for traditional and proposed method are given in Table 4.1. In Table 4.2 column IV illustrates the sensitivities for the optimized (*dbSetup-opt*) method. The run time for computing mismatch sensitivities using different methods are given in Table 4.3.

Table 4.2: Setup Time Sensitivities for Mismatch Variations

Cell	MC Dev(ps)/Error I	*sbMethod* Dev(ps)/Error II	*dbMethod* Dev(ps)/Error III	*dbMethod-opt* Dev(ps)/Error IV
FF1	3.30	3.18 / 4%	3.32 / 1%	3.30 / 0%
FF2	1.86	1.88 / 1%	2.0 / 7%	1.97 / 6%
FF3	3.75	4.14 / 10%	3.7 / 2%	3.45 / 8%
FF4	1.88	1.71 / 10%	1.77 / 6%	1.72 / 9%

From the results, it can be observed that the runtime advantage for global variations using the proposed approach can be as much as $13X$ and on an average is $10X$ compared to the traditional *sbSetup* approach. For mismatch sensitivities the run time improvement using proposed approach, without and with optimization are given in the last two columns of Table 4.3, respectively. It can be observed from these results that there can be a runtime advantage of as much as $30X$ even without pruning of mismatch variables. With inclusion of pruning for mismatch variables, there is as much as $185X$ and on an average $150X$ runtime improvement compared to traditional *sbSetup* approach without much loss of accuracy. Further, it can be observed that for the same sequential cell, the global sensitivities are much smaller than the mismatch sensitivities. This can be understood directly from the Equations 4.4 and 4.5. For global variations, the data and clock path sensitivities have a canceling effect, while for mismatch variations the primary devices contributing to data and clock path sensitivities are different and these sensitivities statistically add up, resulting in

larger mismatch sensitivities.

Table 4.3: Run Time Comparison for Mismatch Sensitivities

Cell	Runtime *sbSetup*	Runtime *dbSetup*	Runtime *dbSetup-opt*
FF1	858.32	30.93 (28X)	6.09 (140X)
FF2	3136	106.26 (30X)	17.1 (185X)
FF3	1323.83	61.32 (22X)	11.2 (118X)
FF4	2163	75.4 (29X)	13.1 (165X)

4.5 Conclusions and Recommendations

We presented in this chapter a fast constraint sensitivity characterization approach. The proposed method relies on measurements for data and clock propagation delays. For characterization of the constraint sensitivities to mismatch variables, we show that the runtime can become infeasible. We presented a runtime optimization method to reduce the number of variables when computing mismatch sensitivities. The method was implemented in an industrial statistical characterization environment and analysis of several 45nm technology flip-flops show that the constraint sensitivity results are within acceptable accuracy of Monte-Carlo simulation results. The results show that, on an average, the proposed approach has $10X$ runtime improvement for global sensitivities and $150X$ improvement for mismatch sensitivities in comparison with the traditional approach.

As discussed earlier, a limitation of the proposed approach is that it requires the internal data and clock paths to be identified for each sequential element. We recommend that a systematic approach be implemented for both annotation of these paths in terms of internal nodes and retention of these nodes appropriately during extraction of the sequential cells. The current work addresses dependence of setup time and clock2q delay. However,

setup time, hold time and clock2q delay are all interdependent. The hold time decreases when the setup time increases and vice-versa. Statistical characterization and statistical timing considering interdependent setup-hold pairs and their variations requires more work and is recommended for future research.

Chapter 5

Statistical Timing Considering Mismatch Variations

5.1 Overview

There are two classes of statistical timing analysis approaches: (a) path-based approach and (b) block-based approach.

- **Path-based Statistical** Timing In a path-based approach, a static timing analysis is run on a design and top few critical paths are selected. The delay of each path is then statistically analyzed using Monte Carlo like simulations. This results in the probability distribution of each path-delay and desired confidence point in the delay distribution is then compared with the target circuit's performance. The advantage of this approach is that it eliminates the problem of delay correlations due to path reconvergence. However, with the number of paths falling within the desired confidence may be very large; the path-based approach can become expensive. The path-based approach also lacks the ability to perform incremental analyses which is imperative for circuit optimizations.

- **Block-based Statistical** Timing Block-based statistical timing analyses target deriving the circuit performance distribution to predict the manufacturing yield. The delay of a circuit is modeled as a random variable and the circuit's performance probability distribution is computed. This inherent need to compute circuit's performance

sensitivity to different process parameters makes statistical timing analysis befitting for circuit optimizations. This approach requires computing the first-order sensitivities for all timing quantities of the circuit with respect to all sources of variation. This provides diagnostics to designers in terms of what sources of variation and provides the diagnostics necessary to improve the robustness of the design.

All references to SSTA in this thesis will be based on block-based analysis algorithms. These algorithms are more efficient and usually have complexity linear in circuit size. There are several challenges in making it a timing sign-off technology; one of them is the *inaccuracies* coming from modeling errors and several assumptions made in the statistical timing algorithms. Few of the factors that impact the accuracy are: (a) linearity and normality assumptions for gate delays, (b) correlations including parametric, spatial and temporal correlations, (c) inaccurate modeling of input slew and output-load variations and its impact on gate delay, etc.

Several SSTA works including [26] [28] [29] [15] [65] have addressed the issue of linearity and normality assumptions. In [30] [14] [32] [31] [57] authors address handling of parametric and spatial correlations using different types of statistical significant component analysis methods. In [57] the authors propose use of separate local random variables for each cell to handle correlations due to path reconvergence. All these SSTA approaches lack accurate consideration of the impact of slew variations on delay and arrival time variations. Both statistical characterization and statistical timing need to account for slew variations. During statistical characterization, each cell/gate need to be characterized for slew variations due to different parametric variations including within-die mismatch/random

variations. Additionally, during statistical timing, the impact of input slew variations and the correlations due to propagation of slew variations need to be modeled accurately.

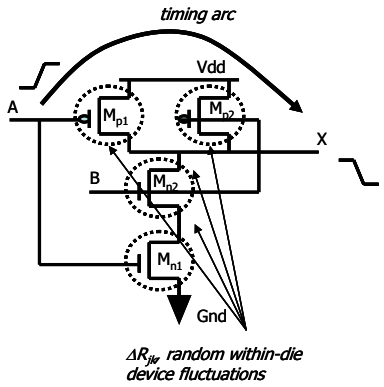


Figure 5.1: Within-Die Variation Results in Fluctuation in Each Device. Variable $\Delta R_{jk} = j^{th}$ Within-Die Parameter, for k^{th} Device

Further, within-die mismatch/random variations result in parameter fluctuations for each device within the cell (illustrated in the above Figure 5.1). Generally these variations due to multiple within-die random parameters (including multiple devices) are modeled using a single gate variable. There are several SSTA techniques proposed to account for both die-to-die (global) and within-die (local) variations. However, these techniques consider the within-die random variations at the cell level and do not account for intra-cell device-to-device mismatch variations. The delay variations of each cell accounting for such intra-cell mismatch variations also need to be included during statistical timing analysis. One approach commonly used to model within-die mismatch variations [24], [38], [37], is to combine the multiple mismatch variables into a single cell-level variable (statistic) during characterization, and keep them as single variable even during propagation of signals in the

timing analysis. This method does not account for any path-reconvergence correlations or slew-based timing correlations. We term this approach as single statistic approach (*SSA*) and show that such a model can result in significant inaccuracy during statistical timing analysis. The error using *SSA* increases with both increase in the logic depth of the design as well as the number of mismatch variables. In [37], an extended timing model was proposed by keeping a separate random variable for each cell. In [57], a fanout based pruning was proposed to account for explosion of these random variables at each node. The authors propose combining multiple local variables to single variable if there is a single fanout. There are two problems with these models: (1) these models do not account for impact of slew variations and the resulting timing correlations and (2) these models consider a single variable for multiple mismatch parameters.

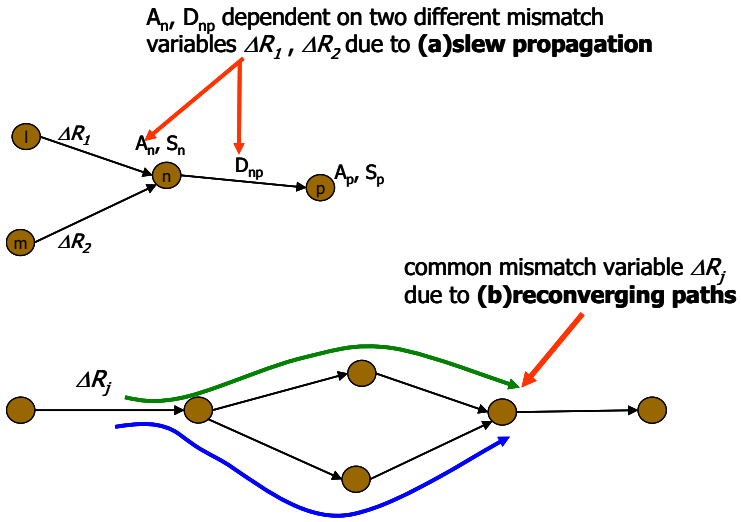


Figure 5.2: (a) Illustration of Slew-Based Correlations and (b) Correlations Due to Reconvergent Paths

Even though mismatch variations are uncorrelated variations, during statistical timing analysis it results in different correlations – these are illustrated in Figure 5.2 and described below.

(a) During timing analysis, propagation of delay sensitivities due to input slew variations result in correlation between input arrival time and the delay computed for each timing arc.

(b) Multiple paths in the data and clock-network share common path-segments. The mismatch variables corresponding to these common-segments result in correlations.

(c) Due to common devices between different timing arcs of the same cell, there are intra-cell delay correlations between timing arcs.

The focus in this chapter is to model the impact of input slew variations and therein, account for slew-based correlations in block-based SSTA. In this chapter, we propose an approach to consider slew variations and its first order effects on delay variations using a common basis of parameter variations. We show that the proposed approach allows for handling correlations due to path reconvergence. Propagation of slew-based correlations in the timing graph results in explosion of variables. We present an efficient pruning technique to handle these large numbers of variables without losing significant accuracy. We use a first-order delay model proposed in the earlier Chapters 3 and 4. The delay models account for both inter-die and within-die mismatch variations.

A simple approach to handle such correlations is to model for each device mismatch variations during statistical characterization and retain every mismatch variable during statistical timing analysis. It can however, be easily seen that retaining such mismatch variables

for all cells in a design with multi-million cells can result in an explosion of mismatch variables. In Chapter 3 we have proposed a variance-based clustering approach which addresses reducing these mismatch variables for each physical parameter during characterization. During statistical timing analysis we propose a simple yet efficient pruning method that reduces the number of variables significantly with minimal loss of accuracy.

The chapter is organized as follows. Section 5.2 reviews the timing preliminaries and the *SSA* method using in SSTA and section 5.3 analyzes the impact of input slew variations in the *SSA* approach. Section 5.4 discusses the proposed timing model and presents a method to handle the explosion of variables. Analyses of several circuits and discussions are presented in Section 5.5.

5.2 Background

5.2.1 Timing Analysis Preliminaries

During timing analysis, a circuit is modeled as a directed acyclic graph $G = \{N, E, n_s, n_f\}$, where $N = \{n_1, n_2, \cdots\}$ is a set of nodes in the design and the edges in the graph are represented as $E = \{e_{ij} | \ \forall \ timing\ arcs\ between\ n_i\ and\ n_j\}$. The nodes represent the input/output pins of gates, and the edges represent the delay associated with each timing arc. The graph G is a topologically sorted graph and has a single source node, n_s and a single sink node, n_f. Any design with multiple inputs/sources or multiple outputs/sinks can be transformed to a single-source, a single-sink timing graph by adding a virtual source n_s and a virtual sink n_f connecting to these inputs and outputs respectively. Generally, signals are propagated from the n_s to n_f using a breadth-first algorithm.

Each signal at the n^{th} node is a triplet with components: $\{S_n, A_n, C_n\}$ where S_n,

A_n are the propagated slew and arrival times respectively, and C_n is the effective load capacitance at n. Each edge e_{ij} is characterized by the edge delay, D_{ij}, which is dependent on input slew S_i and output load C_j. The arrival time, A_{ik} is computed as sum of edge delay, D_{ik} and arrival time at the predecessor node A_i. If multiple signals arrive at node k, then A_k is computed as: $A_k = MAX_{\forall i} A_{ik}$. The MAX operator is used when propagating the latest signals in the graph; while a MIN operator is used for performing an early analysis.

5.2.2 Statistical Timing Analysis

For clarity, few notations are described here.

- A_n, S_n is the latest arrival time and latest slew at the n^{th} node, respectively.
- A_{nk} is the sum of delay, D_{nk}, between nodes n and k and the arrival time at n^{th} fanin node, A_n. That is, $A_{nk} = D_{nk} + A_n$.
- S_{nk} is the output slew due to propagation of delay through edge, e_{nk}.

For statistical timing analysis, all timing quantities are represented as a random variable with Gaussian distribution. The latest arrival time distributions at each node are computed using the two basic operators in its statistical form as follows.

- SUM: when an input arrival time A_i propagates through a gate delay D_{ik}, the output arrival time is computed as $A_{ik} = A_i + D_{ik}$. If A_i, D_{ik} are Gaussian, then A_{ik} can be ensured to be Gaussian.
- MAX: when two arrival times A_{ik}, A_{jk} merge at a cell output, the latest arrival time at k^{th} node is given as $A_k = MAX(A_{ik}, A_{jk})$. Since MAX operator is a non-linear operator, a linear approximation is used and discussed next.

The MAX operator is a non-linear operator and does not retain the resulting distribution to be Gaussian. In order to keep the distributions consistent during propagation in the timing graph, it is desired that the resulting latest arrival time distribution has same representation as delay distributions. This is achieved by approximating A_k to a Gaussian distribution, $\widetilde{A}_k$ as

$$\widetilde{A}_k = P(A_{ik} > A_{jk}) \cdot A_{ik} + (1 - P(A_{ik} > A_{jk})) \cdot A_{jk} + r \cdot \Delta R \tag{5.1}$$

where $P(X > Y)$ is the probability of X being greater than Y (also referred as tightness probability in [24]). From Clark's formulation in [25], the first two moments of A_k can be obtained. Using the formulation in Equation(5.1), the first two moments of $\widetilde{A}_k$ can be matched with that of A_k. This moment-matching results in a residual term with sensitivity r and variable ΔR that is local to each cell.

5.2.3 *SSA* Formulation for Statistical Timing

Using the Single Statistic Approach (SSA), the delay variation (dropping the subscripts nk) can be represented as

$$\Delta D = \sum_{i=1}^{m} d_i \Delta X_i + r^d \Delta R^d \tag{5.2}$$

where d_i, r^d is the delay sensitivity due to ΔX_i, ΔR^d, respectively. The variables ΔX_i are m global parameters of variation and ΔR^d is an equivalent random variable. If there are multiple within-die parameters of variation, all these parameters are statistically combined (considering they are statistically independent to each other) to a single gate level equivalent

local variable and this variable is different for each timing quantity. For example, ΔR^d is a single equivalent parameter local to ΔD.

Generalizing the *SSA* representation, any timing quantity, T including arrival time and slew can be given as

$$\Delta T = \sum_{i=1}^{m} t_i \Delta X_i + r^t \Delta R^t \tag{5.3}$$

where t_i, r^t are sensitivities due to ΔX_i, ΔR^t respectively. As mentioned earlier, the variable ΔR^t is an equivalent random variable local to ΔT. That is, if A_n, S_n are represented using the above *SSA* formulation, then ΔR_n^a, ΔR_n^s are random variables local to ΔA_n, ΔS_n, respectively.

5.3 Impact of Input-Slew Variations

In order to understand and analyze for the impact of input slew variations, we performed several Monte Carlo simulations by injecting input slew variations (without perturbing the global and local parameters of variation). We repeated this for different percentage of input slew variations. The simulations were performed for an inverter using a 65nm technology library. Each experiment was performed with 3000 samples and the impact on delay variations and output slew variations was observed.

Figure 5.3 illustrates the results from these simulations. It can be observed that for output rise transition, a 1σ variation of 8% in input slew results in $\sim$ 5% variation in delay and $\sim$ 6% variation in output slew. Similar observations can be made for different types of cells at different nominal input slew and fanout loading conditions. The amount of variation in cell delay and output slew due to input slew variations is comparable to that due to global/local sources of variations. Consequently, in addition to impact of the parameter

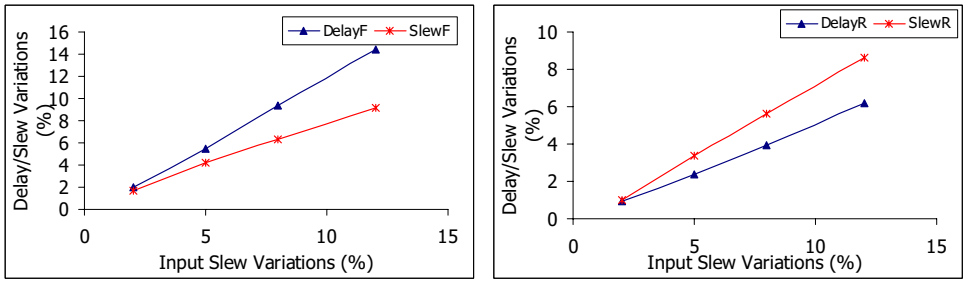
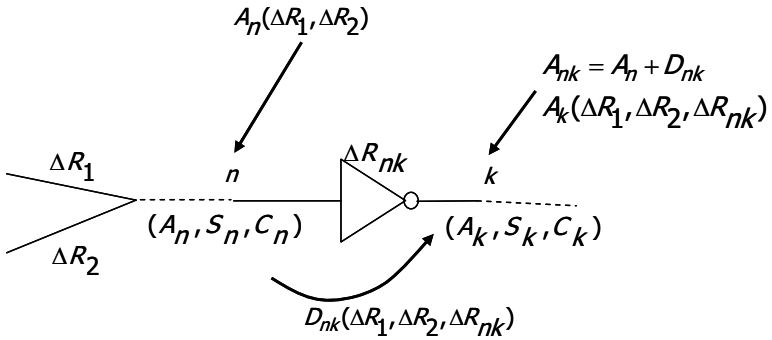


Figure 5.3: Impact of Input Slew Variations

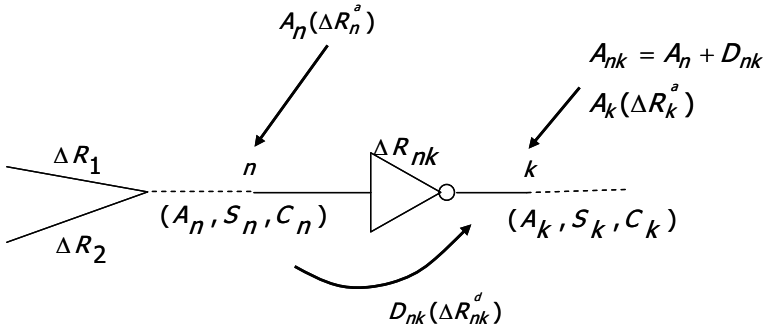
variations on the delay, the variations in input slew also have significant impact on delay and output slew variations. Such an impact cannot be ignored and needs to be accounted during statistical timing analyses.

5.3.1 Slew-based Correlations

Consider a simple inverter with two fanin edges and local variables ΔR_1 and ΔR_2 as illustrated in Figure 5.4(a). Slew is propagated through each edge in the timing graph and the latest slew at the output node is determined. This latest slew is then a function of *all* the within-die variation parameters corresponding to each fanin edge through which it was propagated. For example, the latest slew on the n^{th} node is a function of both fanin variables ΔR_1 and ΔR_2. That is, $S_n = S_n(\Delta R_1, \Delta R_2)$. The latest arrival time at node n is also dependent on these variables, $A_n = A_n(\Delta R_1, \Delta R_2)$. The delay variable, ΔD_{nk} of the inverter is a function of S_n and it has dependence on the parameter variations, ΔR_{nk} within the inverter as well. That is, $\Delta D_{nk} = \Delta D_{nk}(\Delta R_1, \Delta R_2, \Delta R_{nk})$. Thus, now D_{nk} and A_n are correlated; and $A_{nk}, \forall n$ are correlated as well. Even though ΔR_1 and ΔR_2 are independent random variables, due to propagation of the slew, there is a correlation between



(a) Illustration of Slew-Based Correlations



(b) *SSA* Representation

Figure 5.4: Illustration of Problem in Using *SSA* for Slew-based Correlations

S_n and A_n. We term such a correlation as slew-based correlation.

Let us look at the *SSA* representation of this inverter case (Figure 5.4(b)). In this case, the latest slew S_n and arrival time, A_n have random variables ΔR_n^s and ΔR_n^a respectively which are local to the timing quantities. The variables ΔR_n^s and ΔR_n^a are considered to be statistically independent. This implies that the correlation between S_n and A_n due to the common fanin variables ΔR_1 and ΔR_2 is lost. Further, when D_{nk} is computed as a function of S_n, there is no correlation between D_{nk} and A_n. This is happening because of the representation of delay and slew during characterization as a combined statistic for all within-die parameters. During timing analysis, in the *SSA* approach when computing $A_k = MAX_{\forall n}(A_{nk})$ ($S_k = MAX_{\forall n}(S_{nk})$), an equivalent local variable ΔR_k^a (ΔR_k^s) is computed. The result is, ΔR_a^k and ΔR_s^k are now treated as independent random variables.

5.4 Proposed Timing Model

During timing propagation, *SSA* retains the within-die component as a single independent, random variable local to each timing arc. Use of such single variable results in not accounting for slew-based correlations due to within-die mismatch variables, and hence results in significant error in computation of arrival time sensitivities at each node. In the following section, we first propose the generalized multi-statistic approach (*MSA*) and then derive the formulations to include impact of slew variations. A separate random variable is assigned to each timing arc with respect to each within-die variation parameter. Using a common basis of random variables retained in the propagated slew and arrival times, the slew-based correlations are correctly modeled.

We propose here a timing model by representing a separate random variable for

each within-die variation for every timing arc. Let ΔR_{jk} be random variable for each k^{th} timing arc and j^{th} within-die variation parameter. In such a model, any timing quantity at a node can be given as

$$\Delta T = \sum_{i=1}^{m} t_i \Delta X_i + \sum_{j=1}^{n} \sum_{k \in FI} {r^t}_{jk} \Delta R_{jk} \tag{5.4}$$

where t_i is the sensitivity due to ΔX_i and ${r^t}_{jk}$ is sensitivity due to ΔR_{jk}. The variables ΔR_{jk} include variations from all the the fanin (FI) edges.

Now, consider a small perturbation applied to any parameter of variation ΔX. Then slew variations due to this perturbation on k^{th} node can be represented as

$$\Delta S = s \cdot \Delta X \tag{5.5}$$

Let $\gamma = \frac{\partial D}{\partial S}$ be the derived sensitivity of delay with respect to small perturbation in input slew. Multiplying Equation(5.5) with γ, gives the following relation.

$$\Delta D|_S = \gamma \cdot s \cdot \Delta X \tag{5.6}$$

$$\Rightarrow \Delta D = \Delta D|_S + \Delta D|_{\Delta X} \tag{5.7}$$

In Equation(5.7), the delay variation ΔD is the sum of variation due to input slew variations, $\Delta D|_S$ and variation due to each parameter ΔX, $\Delta D|_{\Delta X}$. These two components of delay variation have a common random variable, ΔX and hence, results in additive sensitivities. Generalizing Equation(5.7) for all parameters of variation ΔD can be represented as follows,

$$\Delta D = \sum_{i=1}^{m}[d_i + \gamma \cdot s_i]\Delta X_i + \sum_{j=1}^{n}\sum_{k}[d^p{}_{jk} + \gamma \cdot s^p{}_{jk}]\Delta R^p_{jk} + \sum_{j=1}^{n}\sum_{k} d^q{}_{jk}\Delta R^q_{jk} \qquad (5.8)$$

where ΔR^p_{jk} are within-die mismatch variables due to fanin edges and ΔR^q_{jk} are variables due to arcs in the gate for which delay variation is being computed. The equations for input arrival time and input slew considering mismatch variables can be represented in a similar form.

The values d_i, s_i and $d^p{}_{jk}$, $s^p{}_{jk}$ can be obtained directly by referencing into the statistical characterization results. However, γ needs to be computed on the fly during timing analysis; this is because its value depends on the nominal delay and nominal slew values for each timing arc.

Generalizing above formulations, any timing quantity during statistical timing analysis considering within-die mismatch variations in the proposed model is represented as

$$\Delta T = \sum_{i=1}^{m} t_i \Delta X_i + \sum_{j=1}^{n}\sum_{k \in FI} t_{jk}\Delta R_{jk} \qquad (5.9)$$

where FI = set of all variables in the fanin cone. We term this proposed timing model as multi-statistics approach (*MSA*). The variations in output arrival time, A_{nk} can be easily determined as sum of $A_n$and D_{nk} by retaining variables for each fanin edge. Further, if multiple edges merge at the output, the output arrival time $A_k = MAX_{\forall n} A_{nk}$ is computed using union of mismatch variables from A_{nk}. The linear approximation described in section 5.2.2 is used for the statistical MAX operator. The latest output slew is computed using similar formulations.

5.4.1 Estimation of Error due to *SSA*

The error in *SSA* is a result of not accounting for propagation of slew and the slew-based correlations for the within-die variables. This can be obtained by examining the covariance, C_{DA} between delay variation, ΔD and arrival time variation ΔA using the *MSA*. C_{DA} is given as

$$C_{DA} = \sum_{i=1}^{m}[d_i + \gamma \cdot s_i] \cdot a_i + \sum_{j=1}^{n} \sum_{k \in FI} \gamma \cdot s_{jk} \cdot a_{jk} \tag{5.10}$$

The error due to *SSA* is due to not accounting for the second term in 5.10. Let us simplify the input slew sensitivities, s_{jk} and the arrival time sensitivities a_{jk} to be same for all timing arcs, say s and a respectively. The error due to *SSA* for a timing path can be estimated as

$$\epsilon = n \cdot p \cdot (\sum_{i=1}^{L} \gamma^i \cdot s) \cdot a \tag{5.11}$$

where n is the number of within-die variation parameters, p is an average number of timing arcs within each level in the timing graph, and L is the number of topological levels in the timing graph. Thus, the error due to *SSA* increases with number of fanin edges p, and has a geometric progression of γ with respect to topological levels L.

5.4.2 Handling Variable Explosion

As seen in Equation(5.11), the number of within-die variables in *MSA* increases during propagation of signals in a timing graph. For multi-million gate designs, this can result in an explosion of number of variables during timing analysis. Let us understand this through a simple example.

Consider the timing graph in Figure 5.5, where each edge is assigned a unique identifier. Let one random variable be used for each edge i.e., *n=1*. To determine the variables at node *n8*, the union of variables from $n7 = \{14, 13\}$ and *n4*={NIL} are considered. In addition, variables from immediate fanin edges {8, 7} are included. Hence, variables at node *n8*={14, 13, 8, 7 }. Similarly variables at *n12*=(1, 2...14}.

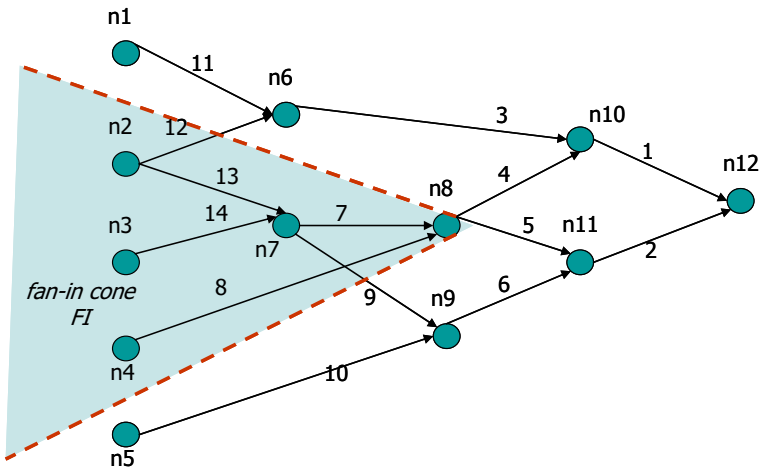


Figure 5.5: Illustration of Propagation of Within-Die Mismatch Variables: Variable Explosion Due to FI Cone in the Timin Graph

The number of variables increases when traversing the graph from source n_s to sink n_f. In order to mitigate this explosion of variables, we perform a simple but efficient pruning of variables without significant loss of accuracy.

We start by assigning a unique index to each edge in the reverse topological order, from sink to source node (see Figure 5.5). During arrival time and slew propagation, the mismatch variables are examined and listed from largest unique index to the smallest. The variables

that have sensitivities smaller than a predefined threshold are eliminated from this list. The pruning is performed only if both arrival time and slew satisfy the threshold. The variables that carry immediate predecessor fanin edge variables are not touched during this pruning. The threshold is determined as a percentage of the effective sensitivity, r due to all within-die variables, where r is given as

$$r = \sqrt{\sum_j \sum_k t_{jk}^2} \tag{5.12}$$

The intuition behind the proposed pruning is that the variables that are significantly further apart have smaller impact on the edges. Propagation of slew results in accumulation of terms as: $\gamma \cdot (1 + \gamma + \gamma^2 + \cdots)$. Generally, γ is a small quantity (< 1) and hence higher order terms coming from edges that are further than immediate predecessors do not have significant impact. Since both arrival time and slew are examined for pruning simultaneously, it allows for keeping a common basis of mismatch variables for arrival time and slew (hence, the correlations are not lost). We observe that this simple pruning approach is very efficient and can result in significant reduction in number of mismatch variables (illustrated in the results section). We term the *MSA* model with inclusion of pruning as *MSA-with-pruning*.

5.4.3 Handling Path Reconvergence

It can trivially be shown that the proposed *MSA* model considering separate variables for each timing arc inherently handles common sub-path in data network. To prove this, it is sufficient to show that variables from a shared common sub-path are propagated to the re-converging node. This is explained with an example. Consider in Figure 5.5 signal propagation paths *P1*{14, 7, 4, 1} and *P2*{14, 7, 5, 2}. Both *P1* and *P2* share the common

sub-path, $SP\{14, 7\}$. To determine the variable list at *n12*, the union of list from *n10* and *n11* are considered. Recursively going back, both nodes *n10* and *n11* include variables from *SP*. Thus, the proposed formulation inherently handles common sub-paths in the data network. Note that while pruning may eliminate a few of the common sub-path variables, the elimination will still retain the variables that are close to the re-converging node. Pruning will only eliminate those variables that have negligible impact on the re-converging node.

5.4.4 Common Segments in Clock Tree

Typically, the circuit's clock network is represented as a clock tree structure with a single source (that represents a clock generator) and multiple sinks (representing registers or flip-flops in the circuit). Then, a depth-first traversal from source to each sink is performed during timing analysis. For statistical timing, the delay distribution of each clock driver is dependent on its input slew variations. The input slew and delay distribution is propagated from the clock tree source to each sink in the clock tree. The delay and slew distributions are computed using the proposed *MSA*. The clock arrival time is then simply the statistical sum of the clock-driver delay distributions from clock source to each sink. The mismatch variables accumulate from clock source to sink and the number of these variables for each clock path depends on the number of clock drivers in that path.

Finally, to determine the performance yield of the design, the clock skew distributions need to be computed between every register pair (if there is a valid data network between the register pairs). The common segment of the clock tree does not contribute to the computation of clock skew distributions. When computing skew sensitivity, the global sensitivities for common segment cancel each other and do not require any additional com-

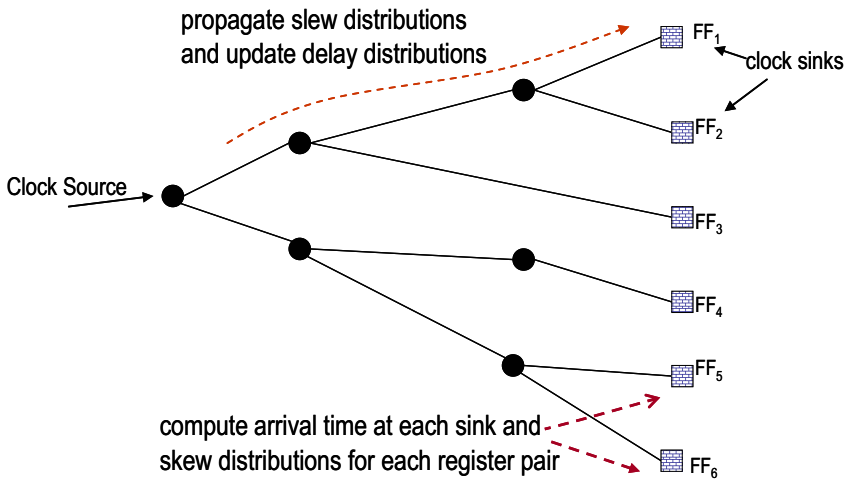


Figure 5.6: Clock Tree Traversal

putation. Using *SSA* requires expensive forward and backward traversals of the clock tree to handle the common segments with respect to within-die variations. However, using *MSA* a single forward traversal of the clock tree is sufficient. The mismatch variables corresponding to each cell along the clock path from source to sink is retained in *MSA* and hence, when computing the skew sensitivity for any register pair the common segment mismatch sensitivities cancel each other. Thus, use of *MSA* allows for handling common clock segments elegantly with no additional traversals or computations required to determine clock skew distributions. More details on clock skew distributions using *MSA* is described in the following Chapter 6.

5.5 Results and Discussion

The statistical characterization algorithms described in Chapter 3 were implemented in an industrial characterization engine. The statistical timing analysis formulations for both *SSA* and *MSA* models were implemented within an industrial gate-level static timing analysis engine. The algorithms were tested on several industry designs. The results are presented for ISCAS benchmark circuits and an industrial design, *DesA*. *DesA* is an SOI 90nm DSP processor core with 45K cells. ISCAS benchmark circuits were synthesized using 90nm technology library. The library cells were characterized for nominal delay/slew and sensitivities using 90nm process technology parameters. To illustrate the improvement in accuracy due to inclusion of mismatch variables, we used a single within-die variation parameter namely, LER and its standard deviation was set at $3\sigma = 5\%$. All experiments were compared with a cell level Monte Carlo simulation considering 3000 samples. In addition to arrival times, the output slew at each node is also tracked during Monte Carlo simulations. The distributions (mean and sigma) for arrival times and slew at each node are obtained.

Table 5.1: Impact of Slew Propagation and Slew-Based Correlations on Arrival Time Variations for Within-Die Variations

	SSA		*MSA*		*MSA-with-pruning*		
Circuit	Mean Error%	Sigma Error%	Mean Error%	Sigma Error%	Mean Error%	Sigma Error%	Variable Reduction
C17	1.4272	22.0712	0.7811	11.1139	0.7811	11.1139	1.1X
C499	0.0100	30.2992	2.0543	3.8309	2.0543	3.8309	2.3X
C1355	-0.6097	36.5005	1.0733	10.8311	1.0733	10.8312	2.4X
C2670	-0.5171	26.0228	-0.4964	-11.8530	-0.4964	-11.8530	7X
C6288	-1.1240	38.7309	0.2220	14.0360	0.2220	14.0365	3.3X
C7552	-0.5543	47.5641	0.7240	10.8989	0.7240	10.8989	2.5X
DesA	-0.7696	16.7725	-0.5362	6.8843	-0.5362	6.8857	3X

We compared both *SSA* and *MSA* formulations. Table 5.1 summarizes the error in arrival

time distributions with respect to the Monte Carlo simulation results for these two methods. The results show that when accounting for the correlations between arrival time and delay accurately using *MSA*, there is significant improvement in accuracy of predicting end point arrival time sensitivities. On average, the absolute error has reduced from $\sim 31\%$ (for *SSA*) to $\sim 9\%$ (for *MSA*).

Further, the pruning algorithm (*MSA*-with-pruning) results were compared with the algorithm without pruning (*MSA*). A very tight threshold of 1% of effective sensitivity was chosen. The results are illustrated in the last three columns of Table 5.1. The table includes the number of variable reduction due to pruning in *MSA-with-pruning*. The results indicate that the error in mean and standard deviation are almost unchanged with pruning of the variables, while there is $\sim 3X$ reduction in number of variables. This indicates that for small circuits there is no significant advantage in pruning; however, $\sim 3X$ reduction in number of variables at each end point of large circuits can have significant advantage on both runtime and storage requirements. Additional reduction in variables may be obtained by setting a more relaxed threshold.

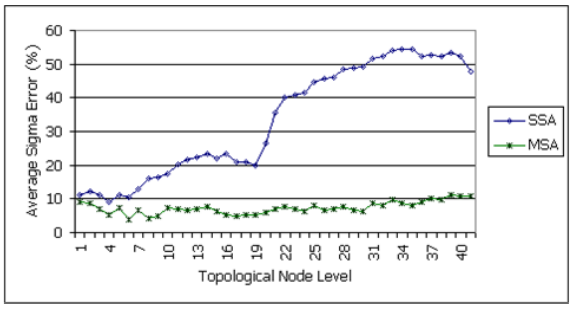


Figure 5.7: *C7552* – *MSA* Vs. *SSA* Level-Wise Average Error in Sigma

To assess the accuracy of the proposed model at each node in the timing graph, we compute an average error at each level in the timing graph as follows: (a) first, the absolute percentage error in standard deviation for each node with respect to Monte Carlo simulation results is computed and (b) then the average of this value for all nodes in a given topological level is determined. Figure 5.7 illustrates this average error in standard deviation for the *C7552* benchmark circuit. It can be observed that the maximum error reduces from $\sim 50\%$ for *SSA* to $\sim 10\%$ for *MSA*. It is interesting to note that the average error is almost constant across the graph for *MSA*; while for *SSA*, the error accumulates and increases at each level as predicted by Equation(5.11). The flat error for *MSA* is the error due to basic assumptions of linearity and gaussian formulations.

5.6 Conclusions

In this chapter, we have presented a model to handle within-die mismatch variations during statistical timing analysis. The statistical timing engine extends the delay model presented in Chapter 3 to account for slew-based correlations during timing propagation. We have developed an efficient pruning algorithm that can handle the explosion of mismatch variables in the timing graph without significant loss of accuracy. The simulation results from timing model considering within-die mismatch variations shows on an average the error reduced from $\sim 31\%$ to $\sim 9\%$.

Chapter 6

Timing Margining Considering Within-Die Clock Skew Variations

6.1 Overview

Chip performance depends on proper clock network design and balancing the arrival times of the clock signal at the latch/flip-flop inputs [66] [62]. A number of factors contribute to clock uncertainty: jitter, variations in the environment of circuit operation (voltage and temperature), and variations in the device and interconnect characteristics resulting from variations in the manufacturing process. The process variations manifest themselves both as global (lot to lot, wafer to wafer, and die to die) variations, systematic (within-die) variations, and local random (within-die) variations in the device and interconnect characteristics. Delay variations in the clock network result in undesired pessimism introduced during static timing analysis that skews the actual timing properties of the circuit. This can result in two incorrect conclusions about the circuit: (a) the actual silicon implementation is operating at a much lower frequency than that predicted by the timing models, and (b) missed functional failure of the circuit. It is straightforward to check for the setup or hold timing violations when considering all delays to be constant [67]. However, due to variations introduced from several sources, delay now becomes a random variable.

Traditional timing sign-off employs a simpler, deterministic, timing analysis. It accounts for global variations by verifying timing correctness of the design at several speed-

corners. The within-die mismatch variations, which are more challenging to handle, are accounted for through certain timing margins that are built into the design, characterization, and analysis procedures. Within-die process variations result in random, mismatch variations between devices. These are not accounted for in the speed corners and can result in unaccounted variations in the clock skew. To account for such random variations, generally designers use a clock uncertainty margin that is set globally for all the timing paths. These margins are designed to be simple and conservative, and are applied globally across the design.

The clock skew variations due to mismatch variations are measurable in silicon and quantifiable during design analysis. In [68] a numerical approach to finding the global clock skew distributions considering within-die variations is presented. On the other extreme, block-based statistical static timing analysis (SSTA) techniques [24] [69] [16] have been proposed to compute the circuit delays as statistical distributions, considering all types and sources of variations. However, a major impediment in the wide-spread adoption of SSTA in the industry is the prohibitively high effort required for statistical characterization of the entire cell library, memories, and other IP blocks. Despite some techniques proposed for efficient statistical characterization, for e.g. [5], the characterization effort is still formidable, especially for characterizing timing sensitivities to local random (mismatch) variational parameters.

In this chapter, we present a method wherein statistical timing analysis is applied only to the clock network to determine realistic (less pessimistic) clock uncertainty margins for random variations, which are then used in the traditional deterministic timing sign-off flow. Since clock network uses only a handful of circuit elements in the library, and statistical timing computations are limited only to the clock network, additional effort needed for

characterization and circuit timing is quite small, but this provides large benefits due to the reduction in pessimism in the margins applied to the clock path delays.

Accurate determination of variation in clock delays (translated to variation in clock skews between pairs of sequential elements) due to random (mismatch) device variations is a key requirement of the proposed approach. However, common sub-paths between clock paths to the launch and capture flip-flops present difficulties in the statistical computation of skew variations [70]. We address this problem through modification of the statistical timing procedure as will be discussed later.

The chapter is organized as follows. Section 6.2 gives an overview of static timing analysis accounting for deterministic clock skew. This section describes the traditional timing methodology that accounts for within-die variations using single constant skew margins. Section 6.3 describes the proposed methodology using statistical analysis of the clock network and thereby, accurately accounting for skew variations. Section 6.4 describes the experimental setup and results observed in a low power application processor. Section 6.5 provides conclusions of the proposed timing flow.

6.2 Preliminaries

6.2.1 Terminology

Launch-Capture Pair (LC-Pair) is the pair of clock pins of two adjacent clocking elements connected by a valid data network (combinational logic) between them. The corresponding clocking elements (flip-flops) are termed as LC-flop pairs.

Local Skew: If (C_i, C_j) are clock arrival times (illustrated in Figure 6.1) at LC-pair (i, j), then local skew is defined as: $S_{ij} = C_j - C_i$

Global Skew is defined for early (late) mode as the upper (lower) bound on clock skew for a given clock domain. Typically, this is computed as the maximum (minimum) of local skews across all LC-pairs. That is, $S_g = \max_{\forall i,j}(S_{ij})$ for early mode and $S_g = \min_{\forall i,j}(S_{ij})$ for late mode.

Branching Point: For a given launch and capture clock path, the point where non-common clock tree segment begins is defined as branching point. *P1*, *P2* (illustrated in Figure 6.1) are two branching points defined for LC-pairs (C_1, C_2) and (C_2, C_3) respectively.

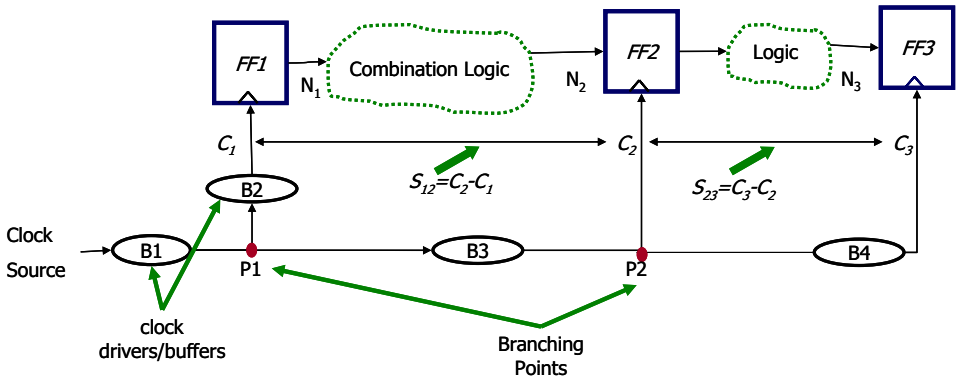


Figure 6.1: Illustration of Launch-Capture Pairs and Branching Points

Without loss of generality, we will assume that the launch and capture elements are clocked by same edge (rising or falling).

6.2.2 Clock Skew and Timing Check

The simplest and most conservative way to accommodate clock skew in static timing analysis (STA) is to use a single, global skew (upper or lower bound) for all LC-pairs. Such global skew is useful for timing checks early in the design flow. However, when the detailed clock distribution network is available, a more accurate local skew is used. Consider LC-pairs (i, j) with local skew S_{ij}. Using local skew between the LC-pairs, the setup slack (U_{slack}) and hold slack (V_{slack}) can be given as follows:

$$U_{slack} = P - U_j + S_{ij} - T_{ij}^{\max} \tag{6.1}$$

$$V_{slack} = T_{ij}^{\min} - V_j - S_{ij} \tag{6.2}$$

where P is the clock period, U_j, V_j are the internal setup and hold constraints due to j^{th}capture flop and $T_{ij}^{\max}$, $T_{ij}^{\min}$ are the latest and earliest data arrival times respectively, between the given $(i, j)LC$-pair. A timing violation occurs when U_{slack} or V_{slack} becomes negative.

Any small variations in clock arrival times may either result in a false timing violation or (worse) result in functional failures in silicon. In order to overcome such violations, designers provide timing margins. Generally, these margins are very conservative and may result in either increasing the number of design optimization iterations or difficulties for the designers to perform any design optimizations.

6.2.3 Current Timing Margining Methodology

Process disturbances are often described by device parameter variations which can be classified into two basic types: *global variations*, which are the same for all devices on the same

chip and *random/mismatch variations*, which vary from device to device. Due to process and environmental variations, clock skew is no more a constant value; but is a statistical quantity. During deterministic STA, variations are accounted using following methods.

- Global/Die-to-die variations: Typically the global variations are embedded as part of the worst-case speed (WCS) and best-case speed (BCS) corner libraries. And, then timing analysis is performed in each such chosen corner.
- Environmental variations: De-rating is a technique where a predefined percentage of cell delay is added to existing cell delay. This technique is typically used to account for delay variations due to voltage (and other environmental) variations.
- Random/Within-die variations: Typically for the clock network a single skew margin is added as clock uncertainty to account for device mismatch variations in the clock network. Additional margin may be provided for clock jitter and other uncertainties in the clock network.

In traditional deterministic timing, a single value M_C is chosen as a pessimistic clock skew margin to account for within-die variations. The value for M_C is generally determined using a predefined percentage of the clock period. Timing analysis is performed by adding the skew margin, M_C to S_{ij} in the timing check equations (described in previous section). Significant pessimism is introduced in the analysis as the margin is pre-determined and the same amount of margin is applied globally for all timing paths. Common path pessimism reduction (CPPR) is achieved by applying the variable component of the margin only to the non-common parts of the launch and capture clock paths [70]. Nevertheless, the pre-defined global margining scheme is still inaccurate since it does not consider the unique sensitivities

of individual paths to variations, which in fact are different due to differences in the cells and routing layers used in different clock paths. Moreover, the constant component of the margin has the effect of adding excess pessimism to launch and capture paths which have lot of sharing.

6.3 Proposed Timing Margining Methodology

The proposed method is described for hold analysis. However, it is applicable to setup analysis also. We confine our approach to a tree topology in clock networks. An overview of the proposed methodology is illustrated in Figure 6.2. It can be described in the following four steps.

Step 1. Pre-characterize the clock-tree cells for delay and slew sensitivities due to mismatch (random) variations using methods described in [5]. The sensitivity characterization is performed at all setup and hold analyses corners.

Step 2. Perform deterministic static timing analysis (DSTA) of the complete SoC at different PVT-corners using the conservative pre-defined single margins. Determine the LC-pairs involved in the timing violations.

Step 3. Perform statistical static timing analysis (SSTA) only on the clock network. The output of this step is the clock skew distributions for all LC-pairs. This step requires identifying the branching point for each LC-pair.

Step 4. Perform margin correction using accurate clock skew variations for each violating LC-pair. Perform an incremental deterministic timing using results from previous step.

Any remaining hold-timing violations are then fixed, and steps 2 through 4 are repeated. Note that, if the hold fixes do not alter the clock network, then there is no need to repeat Step 3.

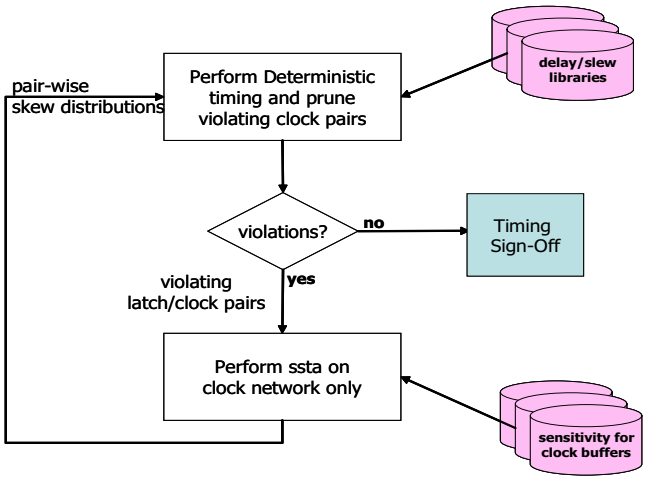


Figure 6.2: Proposed Methodology: SSTA feedback to DSTA

The sensitivity of devices to mismatch variations is different at different timing corners. In order to account for the skew variations arising due to such mismatch variations and correct the hold-time margins, the proposed methodology is applied at each chosen timing corner. We now describe each of the above steps in detail.

6.3.1 Statistical Characterization of Clock Tree Cells

While the traditional timing characterization is done for all the cells in the standard cells library, statistical characterization (that is, characterization of delay and slew sensitivities to variational parameters) is done only for the cells used in the clock network (typically inverters, buffers, and gating cells). Statistical characterization is performed for within-die environmental and device mismatch variations. We model the within-die environmental vari-

ations as systematic components and device mismatch variations as random, uncorrelated variations. The formulation for delay sensitivities is described below.

Let ΔX_i be m number of global/environmental variations. Let n be number of within-die mismatch variations. Let p be number of devices for each clock-buffer. Consider ΔR_{jk} to be a random variable corresponding to each device mismatch variations. The delay of a timing arc considering first order sensitivities to these variations can be represented as

$$D = D_0 + \sum_{i=1}^{m} d_i \Delta X_i + \sum_{j=1}^{n} \sum_{k=1}^{p} \sigma_{jk} \Delta R_{jk} \tag{6.3}$$

where D_0 is the nominal delay value, and is characterized by setting variations ΔX_i, ΔR_{jk} to zero. The quantities, d_i and σ_{jk} are sensitivities of cell delay with respect to ΔX_i and ΔR_{jk} respectively. The values of d_i and σ_{jk} for each cell are obtained from statistical characterization results. Since, ΔR_{jk} are statistically independent random variables and all clock drivers are either inverters or chain of inverters, we simplify above equation as:

$$D = D_0 + \sum_{i=1}^{m} d_i \Delta X_i + \sum_{j=1}^{n} \sigma_j \Delta R_j \tag{6.4}$$

where σ_j is sensitivity due to ΔR_j. If multiple devices are present in the cell, then σ_j is the equivalent sensitivity considering all device fluctuations for j^{th} parameter. This simplification is possible because of a tree structure, which has a single fan-in for each timing arc. If a transistor is multi-fingered as is usually the case in large clock driver cells, the characterization can be simplified by reducing the number of random variables significantly. The method described in Chapter 3 is used for improving runtime efficiency of characterization of multi-fingered transistors.

6.3.2 Full-Chip Deterministic STA

In this step, a gate-level deterministic static timing analysis (DSTA) of the whole design/SoC using the corner libraries is performed. The full-chip DSTA is divided into two phases. In the first phase, a single pessimistic timing margin, M_C, is used to account for within-die variations. LC flop pairs, where at least one path between them has a timing violation, are marked. In the subsequent timing iterations (second phase), DSTA is performed using accurate local skew margins for each LC-pair. This margin is obtained from the statistical clock skew analysis (described in the next section).

Since there are tens of thousands of flip-flops in a design, the number of violating LC-pairs can potentially be millions. To reduce the complexity of SSTA of the clock network, we apply following pruning mechanisms to reduce the number of violating LC-pairs that are passed to SSTA.

- Slack-based pruning: a predefined slack threshold is set to prune the number of violating LC-pairs.
- Degree-based pruning: Capture degree is defined as number of data signals arriving at a given capture flop, FF_c. If this degree is less than a certain threshold, then all LC-pairs that include FF_c are pruned.

6.3.3 Statistical STA on Clock Network

We use the first order statistical delay model described in the previous section (Section 6.3.1) to perform SSTA on the extracted clock tree. The complete clock network including all clock drivers and interconnect parasitics are extracted for the analysis. In order to capture

accurate loading conditions at the leaf (sinks) of the clock tree, all the flops (illustrated in Figure 6.3) are also included during the analysis. SSTA is performed on the clock tree and skew distribution corresponding to each LC-pair is determined.

Skew Variations Local clock-skew is defined as the difference in clock arrival times for a given LC-pair. Clock arrival time in a clock tree is simply the statistical sum of the clock-driver delay distributions from clock source to each sink. Let C_k be the clock arrival time at sink, k in the clock tree. Let q_k be the number of gates for the clock path from clock source to k. The statistical clock arrival time is given as

$$C_k = C_{k,0} + \sum_i \left(\sum_q c_i^q \right) \Delta X_i + \sum_{j=1}^{n} \sum_{p=1}^{q_k} \sigma_j^p \Delta R_{jk}^p \tag{6.5}$$

$$\Rightarrow C_k = C_{k,0} + \sum_i a_{i,k} \Delta X_i + \sum_{j=1}^{n} \sum_{p=1}^{q_k} \sigma_{jk}^p \Delta R_{jk}^p \tag{6.6}$$

where c_i^q, σ_{jk}^p are sensitivities with respect to ΔX_i, ΔR_{jk}^p. These sensitivities are specific to the clock drivers along the path to sink k. Note that all the sensitivity terms for within-die random variables along the path k need to be retained. The statistical clock-skew between any two LC-pairs (k, l) is given as statistical difference between the two clock arrival times, C_k and C_l:

$$S_{kl} = (C_{k,0} - C_{l,0}) + \sum_i (a_{i,k} - a_{i,l}) \Delta X_i + \sum_{j=1}^{n} b_{j,kl} \Delta R_j \tag{6.7}$$

where $b_{j,kl}$ is an equivalent skew sensitivity computed for within-die variable ΔR_j. We will explain how this single equivalent sensitivity can be computed below.

Common segment of clock tree for each LC-pair do not contribute to the clock skew. When computing skew sensitivity, the global arrival time sensitivities for common segment cancel each other without any additional computation. However, for the random,

mismatch components of the skew sensitivities, the random variables corresponding to each stage along the clock paths for (k, l) LC-pairs need to be retained. When computing the final mismatch skew sensitivity for the (k, l) pair the common segment sensitivities cancel each other. Thus, computation of skew mismatch sensitivities requires retaining random variables for each driver in the clock tree. This can be expensive in terms of storage. Also, computation of skew sensitivity will require traversing every path from clock source to sink which can be computationally very expensive. However, if only non-common segments are determined, then the variables corresponding to these non-common segments are statistically independent and can be combined to an equivalent single variable (as represented in equation(6.7). In order to obtain only the non-common segments an efficient clock tree traversal is proposed that is explained below.

Clock Tree Traversal The delay distribution of each clock driver is dependent on its input slew variations. The input slew and delay distribution computations need to be propagated from the clock tree source to each sink in the clock tree. The clock skew sensitivities need to be computed between every flop pair. This can potentially run into several million pairs and can become computationally expensive. Further, as discussed earlier, the common segment of the clock tree does not contribute to the computation of clock skew distributions. This requires identifying the non-common segment for each LC-pair. Consequently, we perform two traversals (illustrated in Figure 6.3).

1. Forward Traversal. During forward traversal, the slew distribution is propagated from input to output of each clock driver. The delay dependence on slew is computed only for single preceding level. The delay distribution for each clock cell is computed

considering the input slew variations. The within-die variables for the clock cell and its fan-in are retained for each edge in the clock tree.

2. Backward Traversal. During this traversal, for each identified launch-capture flop pair, only the non-common clock tree is traversed to compute the skew distribution. A list of violating LC-pairs obtained from DSTA (described in section 6.3.2) is used to compute the skew distributions. During backward traversal, starting from a violating flop the arrival time is propagated backwards to the source node. The skew distribution is computed at every branching node. This allows computing skew distribution without traversing the common segments.

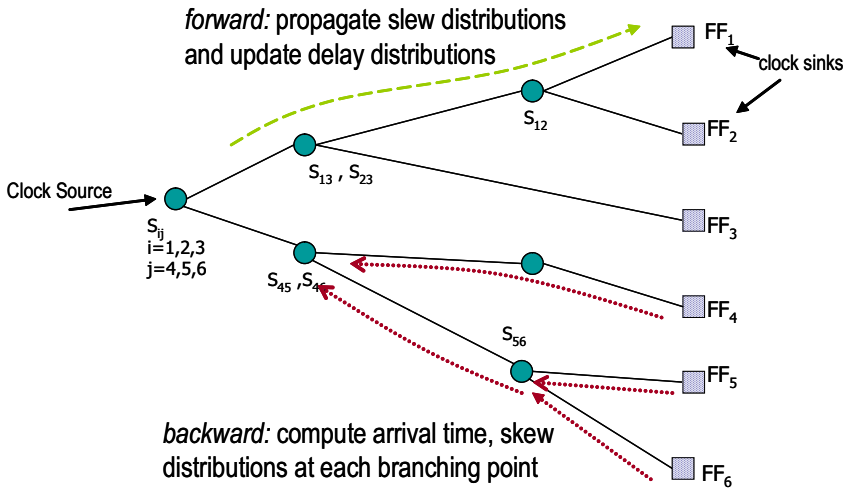


Figure 6.3: Clock Tree Traversal: *forward* performed once, *backward* traversal performed for selected LC-pairs

6.3.4 Accurate Skew Margin Feedback

As mentioned earlier, the skew distributions obtained from SSTA of the clock network are annotated back into the second phase of DSTA. For each LC-pair, a new skew uncertainty

metric is defined using the 3-sigma percentile of skew distribution as $m_{ij} = 3 \cdot \sigma_{ij,r}$. Second phase of DSTA is performed by assigning each violating ij^{th} LC-pair with this metric as new skew margin. Note that the new margin is different for each violating LC-pair. This new margin, m_{ij} is generally much smaller than the original timing margin, M_c which is applied to all LC-pairs. The smaller margin allows for reduction in the number of timing violations.

If there are new timing violations after the margin correction, the timing violations are fixed using design optimizations like buffer insertion, down-sizing certain cells to increase delay along the data path or move the buffers further apart to increase delay through interconnects. Since the number of violations post margin correction is much smaller than original violations, the design optimization iterations are reduced significantly.

If the design optimizations are performed early in the design flow, where there is scope for changes in the clock tree, then the SSTA on the modified clock tree is repeated; and the second phase of DSTA is repeated using new skew margins. However, this case is very rare.

6.3.5 Advantages of SSTA for Clock Tree

A clock tree consists largely of either inverters or buffers. There are several advantages of performing SSTA only on the clock network.

1. Statistical characterization of the complete standard cell library for sensitivities to process variations is one of the most computationally expensive phases for performing SSTA. However, the number of clock drivers in the library is generally much smaller ($\sim 1-2\%$ of the whole library of cells). This allows for significant runtime advantage in performing SSTA only on the clock network.

2. Buffers and inverters do not have same type devices connected in series within the cell. This makes sensitivity/statistical characterization more accurate since no correlation between devices need be considered.

3. SSTA considers a linear approximation of the MAX/MIN operator [24] to get an equivalent Gaussian distribution at the output of each cell. However, clock drivers are single input cells and do not require MAX/MIN operation. This makes the block-based SSTA more accurate for clock tree structures.

4. Clock network is a tree-like structure and hence, correlation due to re-convergent paths need not be considered.

Thus, block-based SSTA allows for more accurate statistical analysis of the clock tree structures. Computationally this method is several orders of magnitude faster than transistor-level spice simulations. The details of the proposed timing margining methodology is illustrated in Figure 6.4.

6.4 Results and Discussion

The proposed margin correction methodology was implemented within our timing flow and test on a low-power application processor platform in 65nm technology. Table 6.1 gives detailed statistics of the platform analyzed.

Clock drivers were characterized for delay and slew sensitivities to threshold voltage (Vt) mismatch variations. A 3-sigma value of $\sim$ 10% was used for Vt variations. Mismatch variations due to nMOS devices and pMOS devices were obtained separately. These characterizations were performed using the targeted $65nm$ technology process node.

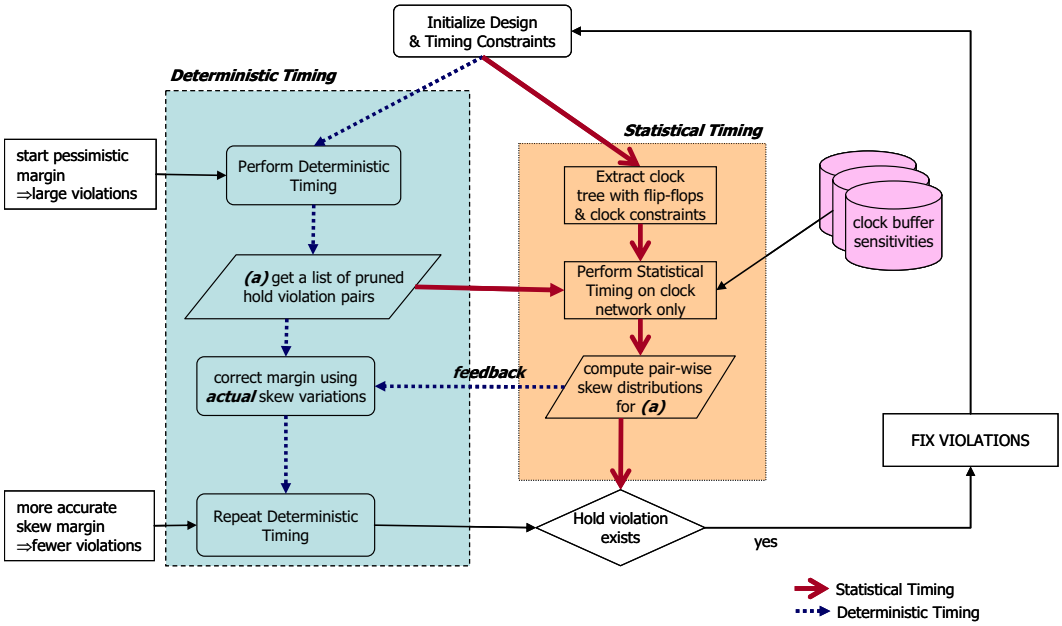


Figure 6.4: Detailed timing margining methodology for skew-margin corrections to account for mismatch variations

Table 6.1: Statistics of the Low Power Processor Platform

Platform Statistics	**Value**
# Gates	~ 429000
# Clock Domains	11
Frequency, Voltage	$\sim 415MHz$ @1.1V
Process corner illustrated	WCS
Initial Margin, M_C	$50ps$
# Violating LC-pairs before margin correction	$\sim 43,000$
# Violating LC-pairs after margin correction	$\sim 4,900$

DSTA was performed using 50ps skew margin (which has been the practice before the proposed methodology to account for random, within-die variations). This was set as initial

margin, M_C or all LC-pairs. This resulted in $\sim 43,000$ LC-pairs violating timing.

Figure 6.5 illustrates the 3-sigma skew variations (on the z-axis) for the violating LC-pairs. On an average the margin required for mismatch variations was reduced from 50ps to 15ps ($\sim 67\%$ reduction). Using the improved margins, the number of violations was reduced from $\sim 43,000$ to $\sim 4,900$ LC-pairs during the second phase of DSTA. This is approximately $10X$ reduction in the number of violations. Moreover, all remaining violations had very small slack violations and were fixed using a single optimization iteration. Effectively, this can be considered to have reduced the number of design optimization iterations by $\sim 10X$.

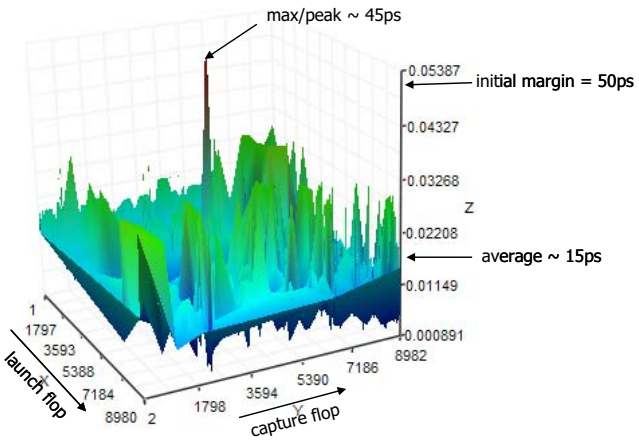


Figure 6.5: Skew Variations for Violating Pairs

We also investigated skew variations for two cases: (a) balanced and unbalanced clock-tree structures and (b) skew variations with and without accounting for common clock-segments. The results for these are discussed below.

6.4.1 Balanced vs. Unbalanced Clock Tree

Even though the mean arrival times for the balanced LC-pairs cancel each other resulting in a zero mean skew, the skew mismatch variations do not cancel each other. Due to random distributions of arrival times for each sink, the variances add up for each LC-pair resulting in always a non-zero mismatch skew variation. The results for few balanced (type B) and unbalanced (type U) LC-pairs are given in Table 6.2. It can be observed that for unbalanced clock tree, the mismatch variations in clock skew is generally smaller than that in balanced tree structure – this is primarily because there are more common segments for the balanced structures. In addition to mismatch variations, an unbalanced tree structure incurs an additional penalty due to unequal sensitivities to environmental variations (eg., Vdd variations).

Table 6.2: Skew Variations: Balanced vs. Unbalanced Clock Tree

LC-Pair	Type	Skew Mean (ps)	Skew Vdd Sigma (ps)	Skew Mismatch Sigma (ps)	Total Skew Sigma (ps)
LC_1	B	0.0	0.0	15.3	15.3
LC_2	B	0.0	0.0	15.1	15.1
LC_3	U	2.6	6.2	2.6	6.7
LC_4	U	11.7	0.3	6.4	6.4

6.4.2 Handling Common Segments

As discussed in earlier sections, handling common segments in the clock network is the most computationally expensive step. It can be observed that, for skew-mean and skew-global-variations, the arrival times from the common segments cancel each other when computing skew distribution. However, for mismatch variations, only the non-common segments need to be accounted when computing arrival times for each LC-pair. We analyzed the method-

ology to identify the amount of pessimism introduced due to inclusion of common-segments when computing skew mismatch variations. This is to see if the backward traversal (which is computationally expensive traversal) can be eliminated. The results are illustrated in Table 6.3. It can be observed that the pessimism introduced due to common segments is very large and can result in significant error in computing the skew margins. Hence, the skew mismatch variations should include only non-common segments.

Table 6.3: Pessimism in Skew Mismatch Variations Due to Common Clock Tree Segments

LC-Pair	Skew-Mismatch w/ common segment (ps)	Skew-Mismatch w/o common segment (ps)	Pessimism due to common segment (ps)
LC_1	13	5.4	1.4X
LC_2	12.3	3.9	2.15X
LC_3	12.7	1.4	8X

6.5 Conclusions and Recommendations

We have presented a timing margin correction methodology that can be easily integrated into a traditional deterministic timing sign-off flow. The proposed methodology leverages the efficacy of block-based statistical analysis when applied to only the clock network. We also presented pruning techniques to reduce the number of LC-pairs for detailed computation of skew distributions. In order to account for shared clock network segments between the launch and capture clock paths, we presented modifications to the statistical timing analysis method. Finally, we presented benchmark results on a low power application processor demonstrating $\sim 67\%$ reduction in the margin and $10X$ reduction in reported timing violations.

Chapter 7

Criticality Metrics for DFM Optimization of Standard Cells

7.1 Overview

Aggressive process scaling has resulted in significant challenges to designs due to process variations and reduced parametric yield. Prior to 65nm designs, complying with minimum design rules (mDR) was sufficient to ensure acceptable yields. With recent challenges in process technologies, there are several recommended design rules (rDR) that is provided for increasing yield. Figure 7.1 illustrates the classification of design rules. The rDR can be divided into two categories: (a) rDR for improving functional yield by decreasing the probability of defects (an example is redundant contact/via insertion) and (b) rDR to reduce variability in critical dimensions. The goal of (b) is to make the designs more robust to certain systematic variations and improve parametric yield.

Standard cells are basic building blocks for digital designs. By applying design rules to the standard cell layouts, design-level or SoC-level compliance to the technology design rules globally can be achieved. So, any small changes made to reduce variability in standard cells can result in significant improvements to design-level or SoC level parametric yield. In [71] a timing-aware cell layout de-compaction was proposed with an objective to minimize critical area. In [72] a timing aware redundant contact insertion was presented. These approaches address the functional yield aspects; however, parametric yield is very

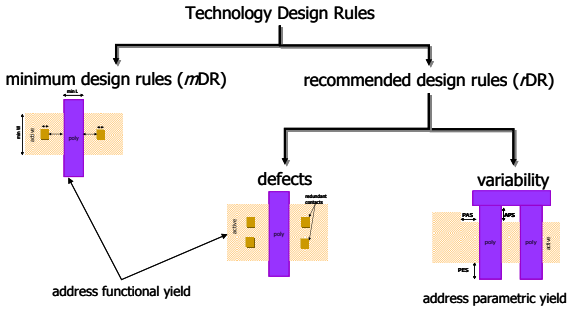


Figure 7.1: Classification of Technology Specific Layout Design Rules

important for designs in 45nm and below technologies.

In addition to the default mDR, there are several rDR specifically poly-endcap, active to poly corner, poly to active corner, etc., that impact the intrinsic devices and are identified as critical parameters for design. Generally rDR for increasing parametric yield is applied through cell layout optimizations. The primary constraints imposed on the cell during such optimization are the area, routability and design rule constraints. These optimizations are applied to all device geometries/layouts within the cell. However, process variations result in each device within the cell to exhibit different variations. Further, the cell's delay sensitivity to these variations differs from one device to the other. During cell optimization there is no knowledge of how variations in each device impacts the cell's delay and leakage variations. There is a growing demand to optimize the cell layouts with a prior knowledge of these performance variations (that impact the parametric yield).

In this chapter, we first define two new metrics from basic delay variation equations. We then investigate the significance of these metrics for cell layout optimizations.

We propose an early estimation of these metric that has good correlation with extracted standard cell layouts. The criticality-aware optimization exploits the within-cell device criticality information to perform selective DFM optimization. The proposed approach takes advantage of devices that are less sensitive to process variations in the cell and relaxes some of the design parameters; while for most critical devices, the rDR as well as systematic layout guidelines for reducing sensitivity to process variations are applied more rigorously. The objective of the proposed optimizations is to enhance existing standard cell layouts for improved parametric yield or reduced variability with minimal penalty on nominal delay and area.

The chapter is divided into the following sections. Section 7.2 describes the preliminaries for standard cell layout design and design-for-manufacturability (DFM) guidelines used for cell layout optimizations. Section 7.3 presents the proposed metrics in two parts: first the within cell criticality metric is described and second the total-sensitivity index metric is presented. The chapter also describes use of these metrics to drive several layout guidelines and optimizations. The experimental setup and results for 45nm technology cells are discussed in Section 7.4.

7.2 Standard Cell DFM Optimization

In this section we describe the typical standard cell layout, optimization and characterization flow that is used across the industry. Following is the terminology or abbreviations used throughout this chapter.

7.2.1 Terminology

- rDR: A set of layout rules that specify the spacing and dimensions for different mask layers based on prior knowledge of manufacturing issues. These guidelines are recommended while the minimum design rules (mDR) are mandatory.

- DFM Score: A weight given to each DFM guideline based on the severity of the guideline to manufacturability.

- DFM Optimization: Layout changes that apply the rDR without violating any design rules. During these optimizations, a typical objective is to not impact the performance of standard cells.

- LPE (Layout Parasitics Extraction): LPE is the step where extraction of parasitics (resistance and capacitance of different layers) from the layout is performed. For standard cells the parasitics are extracted after cells are synthesized from schematic or optimized from an existing layout.

7.2.2 Recommended Design Rules

The recommended layout rules or guidelines address the systematic manufacturing issues. The rDR exists for several layers. However, we examine here only the critical rules that impact the intrinsic device parameters. The layers that form a device or channel region is defined by the poly and active layers. Following are example rDR for layers that form a device (illustrated in Figure 7.2):

- Poly-Endcap Spacing (PES): If the poly endcap is short, then the channel region that is formed near endcap becomes more sensitive to variations. By providing enough

endcap length, such variations can be reduced.

- Active to Poly corner Spacing (APS): Poly corners within a standard cell are generally formed due to contact landing on the poly and/or due to local poly connects (example for multi-fingered transistors). If this spacing is small, then the gate-region has non-rectangular region near the active edge.

- Poly to Active corner Spacing (PAS): Active corners or bends in active form whenever there are two devices with different width adjacent to each other. If the spacing between the active bend to the gate-region is small, then it results in non-rectangular active regions that can cause electrical variations.

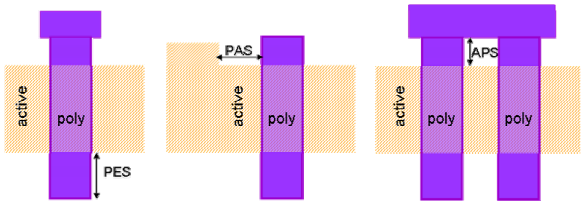


Figure 7.2: Critical Spacing for rDR

Based on prior knowledge (or through silicon characterization), the recommended rules are binned into several levels, $L_1, L_2, \ldots, L_n$. These levels are based on prior knowledge or learning of systematic variations through silicon characterizations. These levels are indicators of severity in terms of geometry variations: a lower bin indicates higher severity while a higher bin indicates a safer region and hence less variability. For the current discussion, we restrict the number of bins to 3, that is L_1, L_2, L_3. For each rule, r and each level, L_i, a

score, C_i^r is assigned such that, $C_1^r > C_2^r > C_3^r$. The score indicates a weight for violating a bin. For example, violating a lower bin is more severe and so carries a higher score. The goal during a DFM optimization is then to minimize the total score in a given standard cell.

7.2.3 Current DFM Optimization Approach

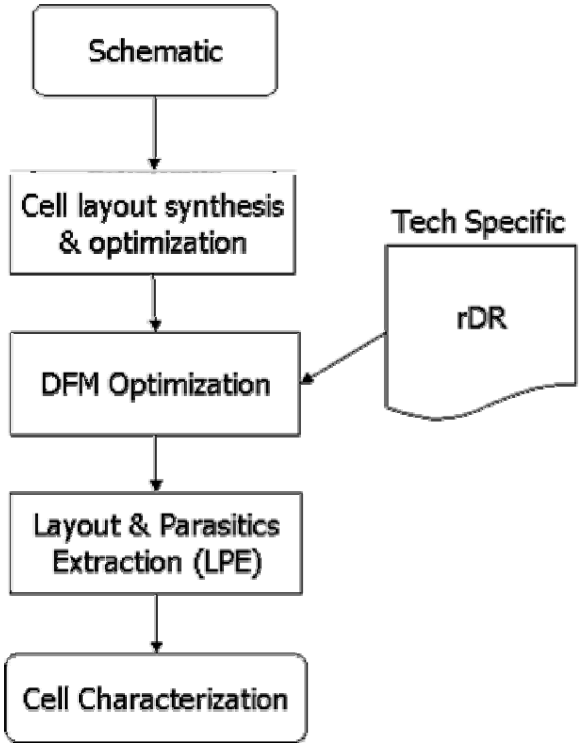


Figure 7.3: Current Cell Synthesis and Optimization Approach

Typically the standard cells are designed with a fixed cell height and the width of the devices derived based on required drive strength. Once the cell height and width are fixed, the cell synthesis and layout optimizations are performed such that total cell area is minimized. Since the number of input and output pins are fixed for a given cell functionality, maximizing routability for these pins is an important criteria during this optimization. Fig-

ure 7.3 illustrates a typical flow that is used currently for standard cell layout synthesis and optimization. A typical flow that is used currently for standard cell layout synthesis and optimization is described below. From a given schematic and architectural specifications of a standard cell, the cell layout is synthesized. At each step of synthesis and optimization, the minimum design rules for all the layers embedded within the standard cells are verified. Before characterization of these standard cells, the cells go through a detailed layout and parasitics extraction phase. The result of this phase is that the internal nets of the cells considering all the layers within the cell are extracted for R, C (parasitics). Additionally, all the devices are also extracted with accurate source / drain area and resistances. The following details the steps in this flow.

- ***Architecture Specification:*** This step involves identifying the optimal number of tracks for cell height and based on the drive-strength requirements, the width of different devices within the cell is determined. Generally the circuit architecture along with device widths is captured in a schematic (SCH) netlist. Additional specifications including cell height, maximum cell width and routability requirements in terms of number of tracks required for pins are captured separately.

- ***Layout Synthesis and optimization:*** In this step the cell layout is synthesized from the architecture specifications with an objective of minimizing the cell-area for given routability and design rule constraints. This step thus ensures that there is no violation of the technology specific mDR.

- ***DFM Optimization:*** This step takes additional recommended design rules and tries to apply to all geometries and devices in the cell. The output of this step is a final

cell layout that is used for layout-parasitic extraction (LPE) and then, for timing or power cell library characterizations.

In the current flow, DFM optimization is performed by identifying opportunities in the standard cell layout to enforce as many recommended rules as practically feasible. While implementing an rDR, the highest level (i.e. the lowest severity level) implementation is attempted so that the DFM risk score is minimized. However, this is done targeting poly/active polygons of every device in the layout, without regard to the relative criticality of the devices to variations.

Let us revisit the goal of DFM optimization for standard cells: the basic objective is to improve parametric yield or reducing systematic variability in cell delay[1]. However, if the functionality and architecture specifications of the cell are such that, there are few (a set of) devices in the cell that do not exhibit any significant contribution to the systematic delay variations, then any DFM optimization effort on these devices will not help in improving the effective parametric yield. Consequently, there are two issues with the current DFM optimization approach for standard cells.

- The DFM guidelines are applied to all devices and all layers without any criticality (or sensitivity to variations) information
- There is no good mechanism to quantify the improvement due to DFM optimization of the standard cells in terms of its performance

[1] for simplicity of discussion only delay variability is used as the metric; however leakage variability may also be another metric

In the proposed approach, we use the fact that all devices in a cell are not equally critical and so the recommended rules can be applied more rigorously for the high criticality devices and less rigorously for the low criticality devices. As the criticality metric of a device, we consider the sensitivity of the cell's delays (or some other performance metrics) to the variation in that device. The criticality metric will be discussed in more detail later.

7.3 Proposed Criticality-aware Optimization

The basic idea in the proposed approach is to exploit the fact that not all devices within a cell result in same delay sensitivity to given process induced layout variations. The proposed approach involves following steps.

- Sensitivity Characterization: All the cells are characterized for delay sensitivities to variation parameters like gate-length (poly width), gate-width (active width), etc.
- Device Criticality Estimation: Based on the sensitivities for all delay arcs (including constraint arcs for latches/flip-flops), the devices are ranked for their criticality within a cell. Additionally a total sensitivity index for each cell is computed. This is used to perform a Pareto analysis of all cells in the library and rank the cells for DFM optimization.
- Criticality-aware DFM Optimization: A weighted DFM score is obtained using the product of the device criticality information with the score for each DFM bin corresponding to the geometries/layers of each device. Layout optimizations are performed for each cell with an objective to minimize the total weighted DFM score.

Figure 7.4 illustrates the steps in the proposed criticality aware DFM optimization approach. Each of these steps are described in detail in the following subsections.

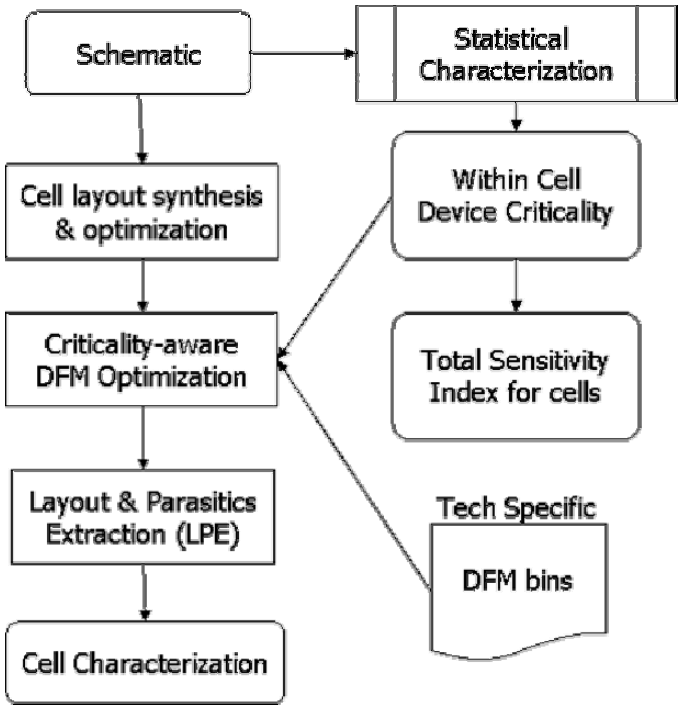


Figure 7.4: Proposed Criticality-aware Cell Optimization Approach

7.3.1 Sensitivity Characterization

Each standard cell in the library is characterized for delay and slew sensitivities to different process variations. Due to process variations each device exhibits a certain variation within the cell. As discussed in earlier chapters, the process variations are broadly divided into (a) inter-die variations and (b) intra-die or within-die variations. From a standard

cell perspective, the within-die variation components result in a device to device random variations.

Let ΔP_i be the variation in each i^{th} device within a cell. Then, the delay sensitivity for each delay arc, α can be represented as follows,

$$\Delta d^{\alpha} = \sum_{i} \sigma_i^{\alpha} . \Delta P_i \tag{7.1}$$

where σ_i^{α} is the delay sensitivity for the delay arc,α due to variations in each device, ΔP_i. By assigning the random variations, ΔP_i, to each device in the cell the sensitivity to delay can be determined using a simple finite-difference approach. As can be seen, if the cell becomes very large or the number of devices within the cell is large, this sensitivity characterization to each device variation may become very expensive. Discussion of characterization to device variations was discussed in previous chapters. As a result of the sensitivity characterization step, the delay sensitivities with respect to each device variation parameter can be determined. The sensitivities for all delay arcs (as well as all constraint arcs) can be determined.

7.3.2 Total Sensitivity Index

Each device variation contributes to all delay arcs; there are some devices which have significant impact on falling arcs while the other devices have significant impact on the rising arcs. In order to understand the contribution of each device with respect to the cell's total performance, all delay arcs need to be considered together. Consequently, we define a new metric, total sensitivity index, Ψ as weighted sum of delay-sensitivities due to all delay arcs in a cell. The total sensitivity index for a cell is given as follows:

$$\Psi = \sum_{\alpha} w^{\alpha}.\Delta d^{\alpha} \tag{7.2}$$

$$\Rightarrow \Psi = \sum_{\alpha} w^{\alpha}.\sum_{i} \sigma_i^{\alpha}.\Delta P_i \tag{7.3}$$

By accumulating all the components of sensitivity due to each device, equation(7.2) can be rewritten as follows,

$$\Psi = \sum_{i} \left(\sum_{\alpha} w^{\alpha}.\sigma_i^{\alpha} \right).\Delta P_i \tag{7.4}$$

$$\Rightarrow \Psi = \sum_{i} \gamma_i.\Delta P_i \tag{7.5}$$

where $\gamma_i = \sum_{\alpha} w^{\alpha}.\sigma_i^{\alpha}$. By doing this, Ψ in equation(7.2), which was initially represented as sum over all delay arcs has been transformed to a representation with the sum indexing over all devices. The total sensitivity index, Ψ now represents a single cell level metric. γ_i represents the total weighted sensitivity of the device variation, ΔP_i, considering all delay arcs within the cell. That is, γ_i is the contribution of i^{th} device to the total sensitivity index of the cell. We term γ_i as *device criticality* metric. Variance and standard deviation of Ψ are represented as $Var\{\Psi\}$ and $\|\Psi\|$ respectively. Above equations are discussed for a single parameter per device. Multiple parameters can be combined to represent as single parameter, by considering statistical independence.

Let us understand the significance of the total-sensitivity index and criticality metric. Consider for example a two input nand cell (NAND2) with four devices $\{N_1, N_2, P_1, P_2\}$. The cell also has four delay arcs: $A(r) \rightarrow X(f)$, $B(r) \rightarrow X(f)$, $A(f) \rightarrow X(r)$, $B(f) \rightarrow X(r)$. For simplicity, consider ΔP_i to be standard normal $N(0, 1)$. Let the weight from each delay arc is same and equal, that is, $w^{\alpha} = 1, \forall \alpha$. This is a good assumption for simple standard cells.

Table 7.1: Device Criticality Metric for NAND2

Device → ↓ Delay Arc	N_1	N_2	P_1	P_2
$A(r) \rightarrow X(f)$	$\sigma^1_{N_1}$	$\sigma^1_{N_2}$	$\sigma^1_{P_1}$	$\sigma^1_{P_2}$
$B(r) \rightarrow X(f)$	$\sigma^2_{N_1}$	$\sigma^2_{N_2}$	$\sigma^2_{P_1}$	$\sigma^2_{P_2}$
$A(f) \rightarrow X(r)$	$\sigma^3_{N_1}$	$\sigma^3_{N_2}$	$\sigma^3_{P_1}$	$\sigma^3_{P_2}$
$B(f) \rightarrow X(r)$	$\sigma^4_{N_1}$	$\sigma^4_{N_2}$	$\sigma^4_{P_1}$	$\sigma^4_{P_2}$
Device Criticality Metric, γ_i	$\sum_\alpha \sigma^\alpha_{N_1}$	$\sum_\alpha \sigma^\alpha_{N_2}$	$\sum_\alpha \sigma^\alpha_{P_1}$	$\sum_\alpha \sigma^\alpha_{P_2}$

For complex cells like flip-flops or latches, the weights represent importance in terms of the criticality of a specific input to output transition. The contributions from each device to each delay arc is illustrated for the NAND2 in Table 7.1. The last row shows the criticality metric of each device to the cell's total sensitivity index. Then, $\|\Psi\| = \sqrt{\sum_i \gamma_i^2}$.

The criticality metric, γ_i, for each device depends on the configuration of the devices within the standard cell, input slew and output loading conditions as well as the parasitics due to intra-cell interconnections (these arise due to different layer geometries including poly, active (diffusion), contact and metal layers).

7.3.3 Total Sensitivity Index for Sequential Cells

Sequential elements [2] are typically characterized using the following timing metrics:

- **Setup Time,** T_{su}: minimum time the data must be stable before the capturing clock edge.
- **Hold Time,** T_h: minimum time the data must remain stable after the capturing clock edge.

[2] we use only flip-flops to discuss the concept for sequential cells; however it can be easily extended to include all types of latches, flip-flops, registers, etc.

- **Clock2q Delay,** T_{clk2Q}: propagation delay from capturing clock edge to valid output transition.

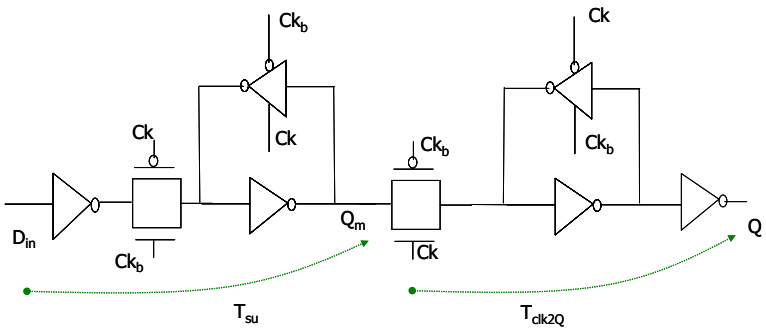


Figure 7.5: Master-Slave Flip-flop: Data (D_{in}) to Output (Q) Timing Arc

For combinational cells, the delay arc captures a timing path from the input transition to output transition. For sequential cells, a timing path from data transition to output transition includes both the delay arc (T_{clk2Q}) as well as the constraint arcs (T_{su}, T_h). The data to output delay, T_{data2Q} considering minimum setup time can be determined using following equation (illustrated in Figure 7.5),

$$T^{\alpha}_{data2Q} = T^{\alpha}_{su} + T^{\alpha}_{clk2Q} \tag{7.6}$$

where, α represents a specific data and output transition pair. The basic sensitivity equation(7.1), for sequential cells can be rewritten as

$$\Delta T^{\alpha}_{data2Q} = \Delta T^{\alpha}_{su} + \Delta T^{\alpha}_{clk2Q} \tag{7.7}$$

By replacing Δd^{α} in equation(7.2) with $\Delta T^{\alpha}_{data2Q}$, the total sensitivity index, and hence the device criticality metric for the sequential cells can be determined. Thus, during statistical characterization, the sequential cells are characterized for both T_{su} and T_{clk2Q} sensitivities. The total sensitivity index and device criticality metric equations remain similar to that described in the previous section. The weighting factor, w^{α} indicates different weights to each data to output transition arcs. The weighting factor may be chosen such that the critical data to output transition arc(s) is set to a high weighting factor, while for the input to output transition arc(s) much smaller weight is assigned.

7.3.4 LPE vs. Schematic Criticality Metric Correlation

Given the criticality metric for the devices within the cell, then the challenge is to determine this metric very early in the standard cell design phase so that the criticality information can be provided during the cell design and layout optimization phases. Typically the "statistical characterization" of standard cells is performed after layout and parasitics extraction. However, for the cell layout and DFM optimization, the device criticality information is required prior to cell layout and parasitics extraction. This becomes a chicken-and-egg problem. To overcome this we performed an exhaustive analysis of the device criticality metric for several standard cells in the library.

We study the correlation between the sensitivities computed using the SCH netlists (these netlists do not have parasitics extracted), γ_i, versus the sensitivities computed with the complete parasitics extracted, γ^e_i. Figure 7.6 illustrates the correlation between γ_i and γ^e_i

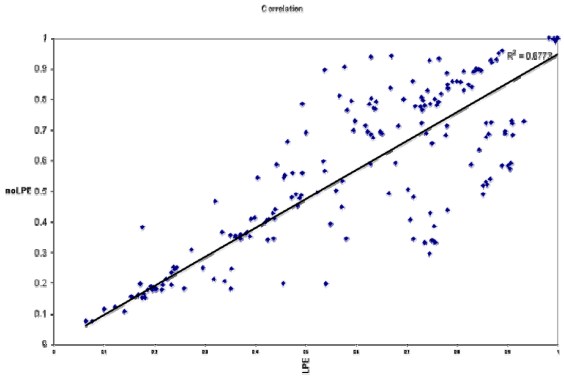


Figure 7.6: Total Sensitivity Index: Correlation Between γ_i and γ_i^e for All Cells

for all (within-cell) devices from $\approx$ 300 standard cells in a 45nm SOI technology library. The correlation factor is ~ 0.7. The correlation factor increases if the cells are categorized into cells with similar input-output pins. That is, consider if the whole library is categorized into (I) all cells with two inputs and one output pin and (II) all the remaining cells, then the correlation between γ_i and γ_i^e is ~ 0.9. This is illustrated in Figure 7.7. The high correlation between these two values indicates that γ_i can be used to guide standard cell layout optimizations. Note that for optimizations, the absolute values of γ_i are not important but the relative values with respect to different devices within the cell determine the criticality of the device. While the schematic criticality metric is used for driving the optimization, the actual delay variations with respect to each device for the SCH and LPE netlists can be different.

7.3.5 Device Rank and DFM Optimization

For simple cells, if the cell configuration is such that all delay arcs have similar or equal delay, then γ_i for all devices should be similar. However, for large cells like custom cells

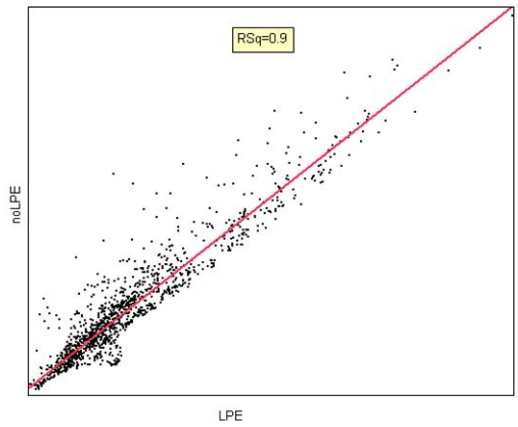


Figure 7.7: Total Sensitivity Index: Correlation Between γ_i and γ_i^e for Cells With Two Input Pins

and flip-flops, there are certain delay arcs that are skewed to have larger delay compared to the other arcs. If γ_i is very distinct for different devices within the cell, then it indicates that there are some devices that are more critical than other devices. That is, there are some devices that exhibit higher cumulative sensitivity to variations; while there are other devices that have smaller cumulative sensitivity to variations. This is a very important input for cell DFM optimizations. If a standard cell is being optimized to reduce systematic variations, then the more sensitive devices can be moved to the DFM-safe regions. Consequently, there are two paradigms to address the problem of layout optimization.

Paradigm *P1*: Synthesize the standard cell layout from scratch to be able to reduce the sensitivity of the most critical devices.

Paradigm *P2*: Start from an existing cell layout and make changes to the layout such that we take advantage of the less critical devices to make the cell more robust to variations.

We define two metrics, namely device rank, R_i and device slack, S_i. These are defined as

follows:

$$R_i = \frac{\gamma_i}{\Gamma} \tag{7.8}$$

$$S_i = 1 - R_i \tag{7.9}$$

where $\Gamma = \gamma_{target}$, a target cumulative device sensitivity for paradigm *P1* and $\Gamma = \max_i \gamma_i$ for paradigm *P2*. Note that S_i can be either negative or positive for paradigm *P1*; however, for paradigm *P2* the value for S_i is always ≥ 0.

The device rank, R_i is 1 for the device that is most critical. The other devices have values ≤ 1 and determine the relative ranking of the devices in terms its criticality within the cell. The intuition behind choosing the device rank to be ≤ 1 is that the rDR are generally defined considering all layout patterns and devices to be equally critical and are all most critical. However, in reality there are several devices within a cell that exhibit smaller contributions to the total delay/performance variations. The DFM scores for any layout changes to these devices should be "attenuated". The proposed criticality metric provides this "attenuation factor".

We define the score for each i^{th} device as C^r_{ji}, where j is the rDR level. For simplicity in notation, from here on we drop the r in C^r_{ji} and represent it as C_{ji}. Each C_{ji} is initialized to C_0 such that, $C_0 \geq C_{1i}$. The DFM optimization problem is then to minimize the total weighted DFM scores. This is given as:

$$\texttt{minimize} \quad \textstyle\sum_i R_i.C_i$$

where $C_i = \sum_j k_j.C_{ji}$ and $\sum_j k_j \leq 1, k_j = 0, 1$. The constraints are the design rule constraints, the cell area constraints and routability constraints for all the input and output pins of cells. Once the critical devices are determined for each cell, we use a greedy approach to DFM optimization. The first step during optimization is to consider only critical devices for applying the rDR such that the devices are closer to the higher DFM level (to minimize the score). In addition to these, the following one or more steps are applied to reduce the cell's DFM risk score that reduces variations in critical devices.

1. Pull the active edge (that is closer to center of cell) such that there are no active bends or notches on either side of the critical devices.
2. Move the poly-contact landing away from critical devices
3. Move the critical devices away from cell boundary. This requires that both nMOS and pMOS devices together are moved away from the cell boundary.
4. Skew drain contacts away from center of device and more closer to the active edge that is closer to center of cell.

For the experiments discussed in the next section, we define a maximum slack for each cell: $S_{max} = \max_i S_i$. For each cell family in the library, the cells are chosen based on larger value of S_{max}.

7.4 Results and Discussion

The proposed criticality aware DFM optimization approach was implemented and evaluated in an industrial 45nm SOI technology library. Several variation parameters including gate

length (L_{eff}), gate width (W_{eff}) and threshold voltage (V_{th}) were used for sensitivity characterization.

We use cells from different families like the NAND, AndOr, XOR, MUX cells to illustrate the approach. A large percentage of the standard cell area in any design is occupied by flip-flops. Even small improvements to making these cells more robust allows for significant gain. Further, we illustrate the approach for sequential cells using a common master-slave flip-flop (MSFF) from a high-performance design.

All the cells were characterized for sensitivities to unit variations in the parameters. The intra-cell device criticality metric and the cell's total sensitivity index were determined from the SCH netlists of these cells.

Table 7.2 shows the results for the combinational cells and the master-slave flip-flop (MSFF). Column I provides the percentage change in the objective, that is the DFM score of the cell by comparing before and after optimization. The results show that on an average the score has reduced by $\approx 34\%$. Columns II and III show the percentage reduction in the nominal cell-delay and the cell's total sensitivity index respectively due to the DFM optimization. All these cells were constrained to zero increase in cell area and zero decrease in input and output pin routability.

The comparison of device ranks using SCH and LPE netlists before optimization for all the cells were performed. As discussed earlier, the correlation between device ranks obtained with and without LPE is very high. The correlation coefficient for MUX cell was ~ 0.9.

The LPE netlists before and after optimization were also characterized for sensitivities. The delay variations with respect to a unit variation in each device, before and after

Table 7.2: Criticality aware DFM Optimization Results

Cell	C_i reduction (%) *I*	Nominal Delay (% increase) *II*	$\|\|\Psi\|\|$ (% increase) *III*
AndOr	30	-0.31	-0.29
MUX	40	0.3	0.01
NAND	34	1.9	1.93
XOR	38	0.81	0.29
MSFF	30	-0.14	-4.78

optimization were compared. Table 7.2 shows the percentage change in $\|\Psi\|$. The delay sensitivities for each device (sorted from left to right, in the order of most critical device to least device) for the MUX cell is illustrated in Figure 7.8. The results show that the delay sensitivities on the most critical device changes from $\sim 11ps$ to $\sim 9ps$, while not penalizing the overall cell delay variation (Table 7.2). The improvement on the most critical device $\approx 18\%$. The weighted DFM score for the cell reduced by 40%.

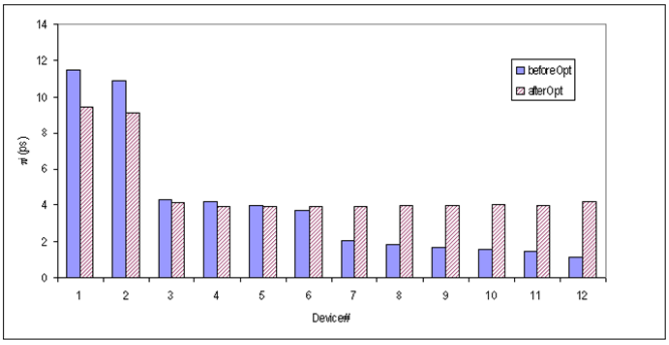


Figure 7.8: Mux: Comparison of Delay Variations Before and After Optimization

The delay sensitivities for each device for the MSFF is illustrated in Figure 7.9.

The results show that the delay sensitivities on the most critical device had reduced by 0.8%; however for the next critical device the reduction is $\sim$ 14%. There is a decrease in the overall cell index (Table 7.2) by 4.78%.

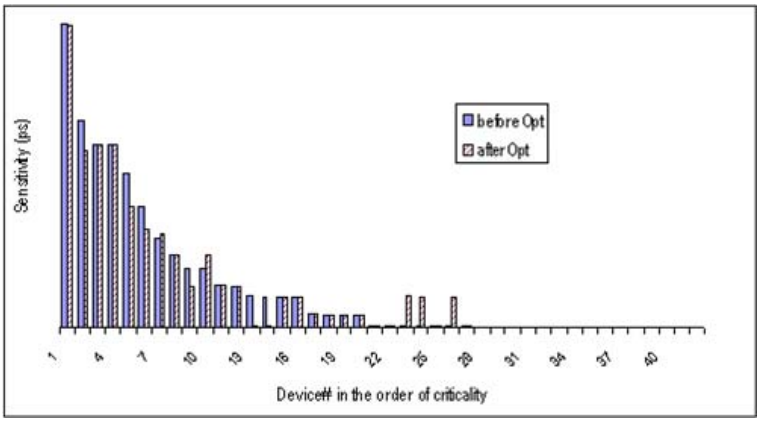


Figure 7.9: Master-Slave Flip-Flop Case: Comparison of Delay Variations Before and After Optimization

Figure 7.10 illustrates one of the layout changes made to the critical devices in the MUX cell. The devices A, B are the non-critical device and critical device respectively. Emphasis was made to apply rDR rigorously for device B. As can be seen, after optimization, the active edge for the critical device, B was pulled closer to cell center. This resulted in two-fold advantage: (a) it increased the PES for the critical device and (b) the APS was eliminated at the side where there is poly-corner due to contact landing. This reduced the DFM score significantly.

Figure 7.11 illustrates one of the layout changes made to the critical devices in the

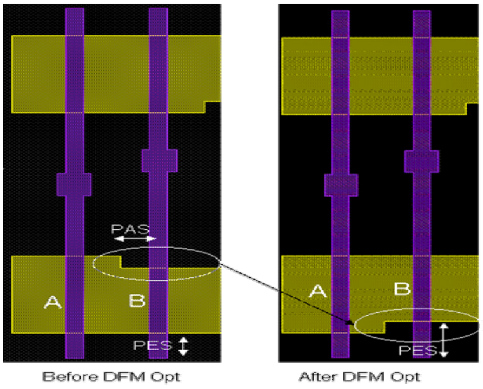


Figure 7.10: MUX Layout Changes Before and After Optimization

MSFF. Here, the critical device, B was moved away from cell-boundary resulting in the critical device to have same poly neighborhoods.

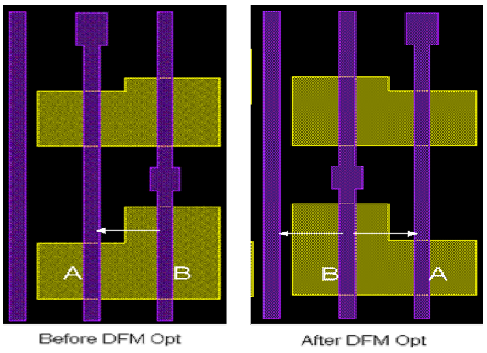


Figure 7.11: Flip-Flop Layout Changes Before and After Optimization

7.5 Conclusions

We presented in this chapter a novel criticality aware standard cell optimization for improving parametric yield. We formulated the problem of cell optimization using a combined device criticality metric and scores defined for recommended design rules. The proposed

method relies on the fact that there are few devices within a cell that are more sensitive to process variations while there are several other devices that have negligible contribution to the delay sensitivities. To account for contributions from all delay arcs, we proposed a total sensitivity index and slack for each cell that allows for ranking the cells across large number of cells in the library for DFM optimization. We implemented the proposed criticality aware DFM optimization approach to pre-optimized layouts for the cells from 45nm SOI technology library. The results show that the proposed approach results in significant reduction in the DFM scores under strict area and routability constraints, without penalizing both nominal delay and the total sensitivity index of the cell.

Chapter 8

Summary and Future Recommendations

The process of statistical characterization starts from silicon characterization, where the sources of variations are characterized into components of systematic and random variations. The results of silicon characterization are captured into spice and circuit models for analysis and optimization. This work has presented new statistical characterization techniques specifically targeting to model the random within-die variations. In Chapters 3 and 4, statistical characterization techniques to handle combinational cells and sequential cells respectively were presented. The proposed techniques can comprehend both inter-die and intra-die variations. It takes advantage of the knowledge of circuit structure and couple it with statistical analysis methods like variance decomposition and significant factor analysis to perform fast statistical characterization of delay and timing constraints with respect to device mismatch variations. Specifically, the within-die random variations result in several intra-cell mismatch variables that require computationally intensive characterizations. Variance based methods are used to determine the significant contributors to delay variance due to mismatch variations and a new clustering approach is proposed for characterization of mismatch sensitivities. In addition to the standard cells being characterized for delay, the sequential cells are characterized for timing constraints like setup and hold time constraints. Generally these constraint characterizations are performed using search-based techniques. The search-based techniques however do not produce consistent measures of

constraint sensitivities. A delay-based method is presented in Chapter4 to compute constraint sensitivity more accurately by accounting for the dependence on delay-degradation. The delay variations due to within-die random variables (mismatch variables) result in a slew-based correlation during timing propagation. In Chapter5 it is shown that the accuracy of statistical timing analysis is improved by accounting for slew-based correlations.

Statistical characterization of circuits for timing models provide a key baseline for understanding the circuit behavior due to different sources of variation. The sensitivity information can also increase yield by reducing the variability during the circuit design itself. In Chapter 7, a standard cell optimization technique is presented that takes advantage of the device sensitivities to different process variations. Based on the device sensitivity information, the optimization method applies recommended design rules selectively on few devices within the cell. This selective optimization results in an improved cell layout for robustness to variations, without penalizing the area and the performance of the cells.

For timing sign-off of any SoC, deterministic static timing analysis is still the standard used in the semiconductor industry. During such analysis, timing margins are added to account for several process and environmental variations. The methodology presented in Chapter 6 takes advantage of SSTA techniques developed specifically for within-die random variations in predicting clock skew variations and providing feedback to deterministic timing analysis. It is presented that statistical timing can be performed more accurately on clock tree and hence, can account for clock skew variations accurately in a timing sign-off flow. Further, an efficient algorithm to traverse the non-common segments of the clock tree to compute skew variations due to the mismatch variables is presented. This novel methodology results in reduction of large number of timing violations and allows focusing more on

the real critical nets for design optimizations.

Going forward, in 32nm technology and beyond, due to introduction of new technologies like double patterning lithography the within-die mismatch variations will continue to increase and be a dominant source of variations. Handling these mismatch variations efficiently for gate timing models and including them in a timing sign-off methodology will be a necessity. Looking into the future, following are specific areas and problems that need further research focus:

- With each technology generation the number of variation parameters are increasing. To generate technology libraries for all such variation parameters can become computationally expensive and many times not required. Statistical parameter selection methods to select performance-oriented (either for timing or for power/leakage) variation parameters is required. These methods need to also target specific circuit styles.

- In addition to setup and hold time dependence on clock-to-q delay, the setup and hold time are also interdependent. This interdependence also need to be handled during characterization of sequential cells for constraint sensitivities.

- Large macros and custom circuits include several combinational and sequential elements. More efficient methods to characterize large macros and custom circuits for variations without losing accuracy need to developed. Methods like adjoint-sensitivity analysis should combine information about the circuits and these need to be addressed within circuit simulators.

- There is more work that need to be done in the area of cell layout optimization considering both systematic and random variations. Lithography is a major contributor

to these variations. Methods to perform lithography-aware standard cell layout optimizations need to be developed. Early cell optimizations by making changes to the transistor width and length that are performance and variability aware need to be developed.

- The recommended design rules are developed to provide a guide for designers to achieve better parametric yield. There is however, no good method to validate each of these design rules for specific circuit styles. Automation of the process of validating each design-rule for parametric yield coupled with circuit information is critical.

In summary this dissertation presents a renewed look at the problem of handling process variations for timing sign-off, which prior to this work were considered to be academically interesting methods but not practical for large designs and methodologies used in the industry. There has been substantial progress made in statistical timing analysis. However, adopting SSTA for timing sign-off in the industry has been very challenging. In order to make statistical timing an acceptable timing sign-off method, the timing methodology that captures variations from silicon and propagates to design analysis need to be simplified. The objective in this dissertation was to give a practical framework towards this end. Statistical methods are considerably more sophisticated and hence newer methods to capture the *statistical nature* of variations in silicon into the traditional *deterministic thinking* of designs need to be developed. The techniques and the methodologies presented in this work will motivate for future research in evolving new analysis and optimization paradigms that can enable designs to be more closer to silicon.

Bibliography

[1] Yun Ye, Frank Liu, Sani Nassif, and Yu Cao. Statistical modeling and simulation of threshold variation under dopant fluctuations and line-edge roughness. In *DAC '08: Proceedings of the 45th ACM/IEEE Design Automation Conference*, pages 900–905, Jun 2008.

[2] P.A. Stolk, F.P. Widdershoven, and D.B.M. Klaassen. Modeling statistical dopant fluctuations in mos transistors. *IEEE Transactions on Electron Devices*, 45(9):1960–1971, Sep 1998.

[3] P.A. Stolk and D.B.M. Klaassen. The effect of statistical dopant fluctuations on mos device performance. *IEDM'96: Technical Digest from the International Electron Devices Meeting*, pages 627–630, Dec 1996.

[4] M. Hane, T. Ikezawa, and T. Ezaki. Atomistic 3d process/device simulation considering gate line-edge roughness and poly-si random crystal orientation effects [mosfets]. *IEDM '03: Technical Digest from the IEEE International Electron Devices Meeting*, pages 9.5.1–9.5.4, Dec. 2003.

[5] Savithri Sundareswaran, Jacob A. Abraham, Alexandre Ardelea, and Rajendran Panda. Characterization of standard cells for intra-cell mismatch variations. In *ISQED '08: Proceedings of the 9th International Symposium on Quality Electronic Design*, pages 213–219, 2008.

[6] A. Srivastava, D. Sylvester, and D. Blaauw. *Statistical analysis and optimization for VLSI: timing and power*. Springer, 2005.

[7] Brian E. Stine, Duane S. Boning, and James E. Chung. Inter- and intra-die polysilicon critical dimension variation. *Proceedings of the SPIE Symposium on Microelectronic Manufacturing*, 2874:27–35, Oct 1996.

[8] Chris Mack and John Robinson. How to characterize critical dimension variations. *A Report by KLA Tencor Corporation*, May 2006.

[9] Micrea Dusa, Hugo Cramer, Ton Kiers, Peter Vanoppen, Jeroen Meessen, Frans Blok, and Stephanie Kremer. A new measure of value for cd metrology tools. *Solid State Technology*, 47(12), Dec 2004.

[10] Jim Bordelon and Prashant Manair. Improving yield through parametric variability characterization and modeling. *Solid State Technology*, 50(11), Nov 2007.

[11] A. Agarwal, D. Blaauw, V. Zolotov, S. Sundareswaran, Min Zhao, K. Gala, and R. Panda. Statistical delay computation considering spatial correlations. *ASP-DAC '03: Proceedings of the Asia and South Pacific Design Automation Conference*, pages 271–276, Jan. 2003.

[12] A. Agarwal, D. Blaauw, and V. Zolotov. Statistical timing analysis for intra-die process variations with spatial correlations. *ICCAD '03: IEEE/ACM International Conference on Computer Aided Design*, pages 900–907, Nov. 2003.

[13] M. Berkelaar. Statistical delay calculation, a linear time model. *Proceedings of ACM/IEEE International Workshop on Timing Issues in the Specification and Synthesis of Digital Systems (TAU)*, pages 15–24, 1997.

[14] Hongliang Chang and Sachin S. Sapatnekar. Statistical timing analysis considering spatial correlations using a single pert-like traversal. In *ICCAD '03: Proceedings of the 2003 IEEE/ACM international Conference on Computer-Aided Design*, page 621, 2003.

[15] Hongliang Chang, Vladimir Zolotov, Sambasivan Narayan, and Chandu Visweswariah. Parameterized block-based statistical timing analysis with non-gaussian parameters nonlinear delay functions. In *DAC '05: Proceedings of the 42nd Design Automation Conference*, pages 71–76, 2005.

[16] Anirudh Devgan and Chandramouli Kashyap. Block-based static timing analysis with uncertainty. In *ICCAD '03: Proceedings of the 2003 IEEE/ACM International Conference on Computer-Aided Design*, pages 607–614, 2003.

[17] A. Gattiker, S. Nassif, R. Dinakar, and C. Long. Timing yield estimation from static timing analysis. *International Symposium on Quality Electronic Design*, pages 437–442, 2001.

[18] Jing-Jia Liou, Kwang-Ting Cheng, S. Kundu, and A. Krstic. Fast statistical timing analysis by probabilistic event propagation. *Proceedings of the Design Automation Conference*, pages 661–666, 2001.

[19] M. Eisele, J. Berthold, D. Schmitt-Landsiedel, and R. Mahnkopf. The impact of intra-die device parameter variations on path delays and on the design for yield of low voltage digital circuits. *IEEE Transactions on Very Large Scale Integration Systems*, 5(4):360–368, Dec 1997.

[20] H. Mahmoodi, S. Mukhopadhyay, and K. Roy. Estimation of delay variations due to random-dopant fluctuations in nanoscale cmos circuits. *IEEE Journal of Solid-State Circuits*, 40(9):1787–1796, Sep 2005.

[21] Yu Cao and Lawrence T. Clark. Mapping statistical process variations toward circuit performance variability: an analytical modeling approach. In *DAC '05: Proceedings of the 42nd Design Automation Conference*, pages 658–663, 2005.

[22] S.Y. Kumar, Jun Li, C. Talarico, and J. Wang. A probabilistic collocation method based statistical gate delay model considering process variations and multiple input switching. *DATE: Proceedings of the 2005 Design, Automation and Test in Europe*, pages 770–775, March 2005.

[23] Seung-Kyum Choi, Robert A. Canfield, and Ramana V. Grandhi. *Reliability-based Structural Design*. Springer, 2006.

[24] C. Visweswariah, K. Ravindran, K. Kalafala, S. G. Walker, and S. Narayan. First-order incremental block-based statistical timing analysis. In *DAC '04: Proceedings of the 41st Design Automation Conference*, pages 331–336, 2004.

[25] Charles E. Clark. The greatest of a finite set of random variables. *Operations Research Vol. 9*, pages 145–162, March-April 1961.

[26] J. Singh and S. Sapatnekar. Statistical timing analysis with correlated non-gaussian parameters using independent component analysis. *DAC'06: Proceedings of the 43rd ACM/IEEE Design Automation Conference*, pages 155–160, 2006.

[27] K. Chopra, B. Zhai, D. Blaauw, and D. Sylvester. A new statistical max operation for propagating skewness in statistical timing analysis. *ICCAD '06: Proceedings of*

the IEEE/ACM International Conference on Computer-Aided Design, pages 237–243, Nov. 2006.

[28] Lizheng Zhang, Weijen Chen, Yuhen Hu, J.A. Gubner, and C.C.-P. Chen. Correlation-preserved non-gaussian statistical timing analysis with quadratic timing model. In *DAC '05: Proceedings of the 42nd Design Automation Conference*, pages 83–88, June 2005.

[29] Y. Zhan, A.J. Strojwas, X. Li, L.T. Pileggi, D. Newmark, and M. Sharma. Correlation-aware statistical timing analysis with non-gaussian delay distributions. In *DAC '05: Proceedings of the 42nd Design Automation Conference*, pages 77–82, June 2005.

[30] Aseem Agarwal, David Blaauw, Vladimir Zolotov, Savithri Sundareswaran, Min Zhao, Kaushik Gala, and Rajendran Panda. Statistical delay computation considering spatial correlations. In *ASPDAC '03: Proceedings of the conference on Asia South Pacific design Automation*, pages 271–276, 2003.

[31] Vishal Khandelwal and Ankur Srivastava. A general framework for accurate statistical timing analysis considering correlations. In *DAC '05: Proceedings of the 42nd Design Automation Conference*, pages 89–94, 2005.

[32] Jiayong Le, Xin Li., and L.T. Pileggi. Stac: Statistical timing analysis with correlation. In *DAC '04: Proceedings of Design Automation Conference*, pages 343–348, 2004.

[33] A. Agarwal, D. Blaauw, V. Zolotov, and S. Vrudhula. Computation and refinement of statistical bounds on circuit delay. *Proceedings of the Design Automation Conference*, pages 348–353, June 2003.

[34] A. Agarwal, D. Blaauw, V. Zolotov, and S. Vrudhula. Statistical timing analysis using bounds [ic verification]. *Proceedings of the Design, Automation and Test in Europe Conference, 2003*, pages 62–67, 2003.

[35] A. Agarwal, V. Zolotov, and D.T. Blaauw. Statistical timing analysis using bounds and selective enumeration. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 22(9):1243–1260, Sept. 2003.

[36] Hongliang Chang and S.S. Sapatnekar. Full-chip analysis of leakage power under process variations, including spatial correlations. In *DAC '05: Proceedings of the 42nd Design Automation Conference*, pages 523–528, Jun 2005.

[37] Lizheng Zhang, Weijen Chen, Yuhen Hu, and Charlie Chung-Ping Chen. Statistical timing analysis with extended pseudo-canonical timing model. In *DATE '05: Proceedings of the conference on Design, Automation and Test in Europe*, pages 952–957, 2005.

[38] Sarvesh Bhardwaj, Praveen Ghanta, and Sarma Vrudhula. A framework for statistical timing analysis using non-linear delay and slew models. In *ICCAD '06: Proceedings of the 2006 IEEE/ACM International Conference on Computer-Aided Design*, pages 225–230, 2006.

[39] Ruiming Chen and Hai Zhou. Clock schedule verification under process variations. *ICCAD-2004: IEEE/ACM International Conference on Computer Aided Design*, pages 619–625, Nov. 2004.

[40] Lizheng Zhang, Yuhen Hu, and Chungping Chen. Statistical timing analysis in sequential circuit for on-chip global interconnect pipelining. *DAC '04: Proceedings of*

the 41st Design Automation Conference, pages 904–907, 2004.

[41] Mango C-T Chao, Li-C Wang, Kwang-Ting Cheng, and S. Kundu. Static statistical timing analysis for latch-based pipeline designs. *ICCAD '04: IEEE/ACM International Conference on Computer Aided Design*, pages 468–472, Nov. 2004.

[42] Lizheng Zhang, Jengliang Tsai, Weijen Chen, Yuhen Hu, and C.C.-P. Chen. Convergence-provable statistical timing analysis with level-sensitive latches and feedback loops. *Proceedings of the Asia and South Pacific Conference on Design Automation*, pages 941–946, Jan. 2006.

[43] Ying Liu, L.T. Pileggi, and A.J. Strojwas. Model order-reduction of rc(l) interconnect including variational analysis. *DAC'99: Proceedings of the 36th Design Automation Conference*, pages 201–206, 1999.

[44] K. Agarwal, D. Sylvester, D. Blaauw, F. Liu, S. Nassif, and S. Vrudhula. Variational delay metrics for interconnect timing analysis. *DAC'04: Proceedings of the 41st Design Automation Conference*, pages 381–384, 2004.

[45] P. Ghanta and S. Vrudhula. Variational interconnect delay metrics for statistical timing analysis. *ISQED '06: 7th International Symposium on Quality Electronic Design*, pages 19–24, March 2006.

[46] D. Sinha and Hai Zhou. A unified framework for statistical timing analysis with coupling and multiple input switching. *ICCAD'05: IEEE/ACM International Conference on Computer-Aided Design*, pages 837–843, Nov. 2005.

[47] M. Agarwal, K. Agarwal, D. Sylvester, and D. Blaauw. Statistical modeling of cross-coupling effects in vlsi interconnects. *ASP-DAC '05: Proceedings of the Asia and South Pacific Design Automation Conference*, pages 503–506, Jan. 2005.

[48] R. Gandikota, D. Blaauw, and D. Sylvester. Modeling crosstalk in statistical static timing analysis. *DAC'08: Proceedings of the 45th ACM/IEEE Design Automation Conference*, pages 974–979, June 2008.

[49] A. van Griensven, T. Meixner, S. Grunwald, T. Bishop, M. Diluzio, and R. Srinivasan. A global sensitivity analysis tool for the parameters of multi-variable catchment models. *Journal of Hydrology*, 324:10–23, Apr 2006.

[50] Karen Chan, Andrea Satelli, and Stefano Tarantola. Sensitivity analysis of model output: Variance-based methods make the difference. *Proceedings of the 1997 Winter Simulation Conference*, pages 261–268, Apr 1997.

[51] G.E.B.Archer, Andrea Satelli, and I.M. Sobol. Sensitivity measures, anova-like techniques and the use of bootstrap. *Journal of Statistical Computation and Simulation*, 58(2):99–120, May 1997.

[52] Andrea Satelli. Sensitivity analysis for importance assessment. *An International Journal for Risk Analysis*, 22(3):579–590, Jul 2002.

[53] Andrea Satelli. Making best use of model evaluations to compute sensitivity indices. *Computer Physics Communications*, 145(2):280–297, Apr 2002.

[54] S. Tarantola, D. Gatelli, and T.A.Mara. Random balance designs for the estimation of first order global sensitivity indices. *Reliability Engineering and System Safety*, 91(6):717–727, Jun 2006.

[55] Scheaffer McClave. *Probability and Statistics for Engineers, Second Edition.* Duxbury, 1986.

[56] Douglas C. Montgomery and George C. Runger. *Applied Statistics and Probability for Engineers, Third Edition.* Wiley, 2006.

[57] Lizheng Zhang, Yuhen Hu, and Charlie Chung-Ping Chen. Statistical timing analysis with path reconvergence and spatial correlations. In *DATE '06: Proceedings of the conference on Design, Automation and test in Europe*, pages 528–532, 2006.

[58] J. A. G. Jess, K. Kalafala, S. R. Naidu, R. H. J. M. Otten, and C. Visweswariah. Statistical timing for parametric yield prediction of digital integrated circuits. In *DAC '03: Proceedings of the 40th Design Automation Conference*, pages 932–937, 2003.

[59] Kenichi Okada, Kento Yamaoka, and Hidetoshi Onodera. A statistical gate-delay model considering intra-gate variability. In *ICCAD '03: Proceedings of the 2003 IEEE/ACM international conference on Computer-aided design*, pages 908–913, 2003.

[60] Kenichi Okada, Kento Yamaoka, and Hidetoshi Onodera. A statistical gate delay model for intra-chip and inter-chip variabilities. In *ASPDAC: Proceedings of the 2003 Asia South Pacific Design Automation Conference*, pages 31–36, 2003.

[61] Ming. Qu and M.A. Styblinski. Statistical characterization and modeling of analog functional blocks. *IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 121–124, May-June 1994.

[62] Vladimir Stojanovic and Vojin G. Oklobdzija. Comparative analysis of master-slave latches and flip-flops for high-performance and low-power systems. *IEEE Journal of Solid-State Circuits*, 34:536–548, 1999.

[63] Shweta Srivastava and Jaijeet Roychowdhury. Interdependent latch setup/hold time characterization via euler-newton curve tracing on state-transition equations. In *DAC '07: Proceedings of the 44th Design Automation Conference*, pages 136–141, 2007.

[64] Safar Hatami, Hamed Abrishami, and Massoud Pedram. Statistical timing analysis of flip-flops considering codependent setup and hold times. In *GLSVLSI '08: Proceedings of the 18th ACM Great Lakes symposium on VLSI*, pages 101–106, 2008.

[65] S. Abbaspour, H. Fatemi, and M. Pedram. Parameterized block-based non-gaussian statistical gate timing analysis. *ASPDAC '06:Asia and South Pacific Conference on Design Automation*, pages 947–952, Jan. 2006.

[66] D. Harris, M. Horowitz, and D. Liu. Timing analysis including clock skew. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 18(11):1608–1618, Nov 1999.

[67] Jindrich Zejda and Paul Frain. General framework for removal of clock network pessimism. In *ICCAD '02: Proceedings of the IEEE/ACM International Conference on Computer Aided Design*, pages 632–639, 2002.

[68] Aseem Agarwal et al. Statistical clock skew analysis considering intradie-process variations. In *IEEE Transactions on Computer-Aided Design, Vol. 23 No. 8*, pages 1231–1242, august 2004.

[69] Chandu Visweswariah. Death, taxes and failing chips. In *DAC '03: Proceedings of the 40th Design Automation Conference*, pages 343–347, 2003.

[70] R. Chen, E.A.Foreman, P.A. Habitz, J.G. Hemmett, K. Kalafala, J.S. Piaget, P. Qi, N.Venkateswaran, C. Visweswariah, J. Xiong, and V. Zolotov. Static timing: Back to our roots. In *International Workshop on Timing Issues in the Specification and Synthesis of Digital Systems (TAU)*, pages 130–136, 2007.

[71] Tetsuya Iizuka, Makoto Ikeda, and Kunihiro Asada. Timing-aware cell layout decompaction for yield optimization by critical area minimization. *IEEE Transactions on Very Large Scale Integrated Systems*, 15(6):716–720, 2007.

[72] Tetsuya Iizuka, Makoto Ikeda, and Kunihiro Asada. Timing driven redundant contact insertion for standard cell yield enhancement. In *Proceedings of IEEE International Conference on Electronic Circuits Systems*, pages 704–707, 2006.

[73] S. Bhardwaj, S. Vrudhula, P. Ghanta, and Y. Cao. Modeling of intra-die process variations for accurate analysis and optimization of nano-scale circuits. In *Proceedings of the 2006 43rd ACM/IEEE Design Automation Conference*, pages 791–796, 2006.

[74] A. Goel and S. Vrudhula. Current source based standard cell model for accurate signal integrity and timing analysis. In *DATE '08: Proceedings of the 2008 Design, Automation and Test in Europe*, pages 574–579, Mar 2008.

[75] Patrick G. Drennan. Understanding mosfet mismatch for analog design. *IEEE Journal of Solid-State Circuits*, 38(3):450–456, Mar. 2003.

[76] B. Cline, K. Chopra, D. Blaauw, and Yu Cao. Analysis and modeling of cd variation for statistical static timing. *ICCAD '06: IEEE/ACM International Conference on Computer-Aided Design*, pages 60–66, Nov. 2006.

[77] Lizheng Zhang, Yuhen Hu, and C.C.-P. Chen. Block based statistical timing analysis with extended canonical timing model. *ASPDAC'05: Proceedings of the Asia and South Pacific Design Automation Conference*, 1:250–253 Vol. 1, Jan. 2005.

[78] Hamed Abrishami, Safar Hatami, Behnam Amelifard, and Massoud Pedram. Nbti-aware flip-flop characterization and design. In *GLSVLSI '08: Proceedings of the 18th ACM Great Lakes symposium on VLSI*, pages 29–34, 2008.

[79] B. Amelifard, S. Hatami, H. Fatemi, and M. Pedram. A current source model for cmos logic cells considering multiple input switching and stack effect. *DATE '08: Proceedings of the Design, Automation and Test in Europe*, pages 568–573, March 2008.

[80] Michael Orshansky and Kurt Keutzer. A general probabilistic framework for worst case timing analysis. In *DAC '02: Proceedings of the 39th Design Automation Conference*, pages 556–561, 2002.

[81] A. Rajaram and D.Z. Pan. Fast incremental link insertion in clock networks for skew variability reduction. *ISQED '06: 7th International Symposium on Quality Electronic Design*, pages 79–84, March 2006.

[82] A. Rajaram and D.Z. Pan. Robust chip-level clock tree synthesis for soc designs. *DAC '08: Proceedings of the 45th ACM/IEEE Design Automation Conference*, pages 720–723, June 2008.

[83] A. Rajaram and D.Z. Pan. Meshworks: An efficient framework for planning, synthesis and optimization of clock mesh networks. *ASPDAC '08: Proceedings of the Asia and South Pacific Design Automation Conference*, pages 250–257, March 2008.

[84] Joon-Sung Yang, A. Rajaram, N. Shi, Jian Chen, and D.Z. Pan. Sensitivity based link insertion for variation tolerant clock network synthesis. *ISQED '07: 8th International Symposium on Quality Electronic Design*, pages 398–403, March 2007.

[85] D. Patel. Charms: characterization and modeling system for accurate delay prediction of asic designs. *Custom Integrated Circuits Conference, 1990., Proceedings of the IEEE 1990*, pages 9.5/1–9.5/6, May 1990.

[86] Mehmet A. Crit and Philippe Hurat. Automated cell library design is ready for the challenge. *White Paper*, Jun 2001.

[87] Mehmet A. Crit. Positional sizing: A recipe for high speed and low power cell libraries. *White Paper*, Mar 2002.

[88] Spiridon Nikolaidis, Er Chatzigeorgiou, and Student Member. Modeling the transistor chain operation in cmos gates for short channel devices. *IEEE Transactions on Circuits and Systems*, 46:363–367, 1999.

[89] Zhaojun Wo and Israel Koren. Effective analytical delay model for transistor sizing. In *ASP-DAC '05: Proceedings of the Asia South Pacific Design Automation Conference*, pages 387–392, 2005.

[90] Robert Boone, Domenico Loparco, Fabio Melchiori, and Matt Thompson. Critical feature and improvability analysis: An effective path to dfm closure. In *White Paper: Mentor Graphics' Technical Publication*, 2008.

[91] Vineeth Veetil, Dennis Sylvester, and David Blaauw. Efficient monte carlo based incremental statistical timing analysis. In *DAC '08: Proceedings of the 45th Design Automation Conference*, pages 676–681, 2008.

[92] Javid Jaffari and Mohab Anis. On efficient monte carlo-based statistical static timing analysis of digital circuits. In *ICCAD '08: Proceedings of the IEEE/ACM International Conference on Computer Aided Design*, pages 196–203, Nov 2008.

[93] Diego A. Alvarez. A monte carlo-based method for the estimation of lower and upper probabilities of events using infinite random sets of indexable type. *Fuzzy Sets System*, 160(3):384–401, 2009.

[94] Amith Singhee and Rob A. Rutenbar. From finance to flip flops: A study of fast quasi-monte carlo methods from computational finance applied to statistical circuit analysis. In *ISQED '07: Proceedings of the 8th International Symposium on Quality Electronic Design*, pages 685–692, 2007.

[95] Tetsuya Iizuka, Makoto Ikeda, and Kunihiro Asada. Opc-friendly de-compaction with timing constraints for standard cell layouts. In *ISQED '07: Proceedings of the 8th International Symposium on Quality Electronic Design*, pages 776–781, 2007.

[96] Raphael Bingert, Alain Aurand, Jean-Claude Marin, Eric Balossier, Thierry Devoivre, Yorick Trouiller, Florent Vautrin, Nishath Verghese, Richard Rouse, Michel Cote, and Philippe Hurat. Implementation of silicon-validated variability analysis and optimization for standard cell libraries. *Proceedings of the SPIE Advanced Lithography 2008 - Design for Manufacturability through Design-Process Integration*, Feb 2008.

[97] Amit Jain and David Blaauw. Slack borrowing in flip-flop based sequential circuits. In *GLSVSLI '05: Proceedings of the 15th ACM Great Lakes symposium on VLSI*, pages 96–101, 2005.

[98] N. Shenoy, R.K. Brayton, and A.L. Sangiovanni-Vincentelli. Graph algorithms for clock schedule optimization. *ICCAD '92: Digest of Technical Papers at the IEEE/ACM International Conference on Computer-Aided Design*, pages 132–136, Nov, 1992.

[99] S. Abbaspour, H. Fatemi, and M. Pedram. VGTA: variation-aware gate timing analysis. *ICCD '05: Proceedings of the IEEE International Conference on Computer Design: VLSI in Computers and Processors*, pages 351–356, Oct. 2005.

[100] R. Gandikota, K. Chopra, D. Blaauw, D. Sylvester, M. Becer, and Joao Geada. Victim alignment in crosstalk aware timing analysis. *ICCAD '07: IEEE/ACM International Conference on Computer-Aided Design*, pages 698–704, Nov. 2007.

[101] R. Gandikota, K. Chopra, D. Blaauw, D. Sylvester, and M. Becer. Top-k aggressors sets in delay noise analysis. *DAC'07: Proceedings of the 44th ACM/IEEE Design Automation Conference*, pages 174–179, June 2007.

[102] V. Kheterpal, V. Rovner, T. G. Hersan, D. Motiani, Y. Takegawa, Andrzej J. Strojwas, and Lawrence T. Pileggi. Design methodology for ic manufacturability based on regular logic-bricks. In *DAC'05: Proceedings of the Design Automation Conference*, pages 353–358, 2005.

[103] R. X. T. Nijssen and C. A. J. Eijk. Eijk. regular layout generation of logically optimized datapaths. In *Proceedings of the International Symposium on Physical Design*, pages 42–47, 1997.

[104] H. Aikawa, E. Morifuji, T. Sanuki, T. Sawada, S. Kyoh, A. Sakata, M. Ohta, H. Yoshimura, T. Nakayama, M. Iwai, and F. Matsuoka. Variability aware modeling and characterization in standard cell in 45nm cmos with stress enhancement technique. *Digest of Technical Papers from Symposium on VLSI Technology*, pages 90–91, 2008.

[105] Noel Menezes, Chandramouli Kashyap, and Chirayu Amin. A "true" electrical cell model for timing, noise, and power grid verification. In *DAC '08: Proceedings of the 45th Design Automation Conference*, pages 462–467, 2008.

[106] Peter Feldmann and Soroush Abbaspour. Towards a more physical approach to gate modeling for timing, noise, and power. In *DAC '08: Proceedings of the 45th Design Automation Conference*, pages 453–455, 2008.

[107] S. Raja, F. Varadi, M. Becer, and J. Geada. Transistor level gate modeling for accurate and fast timing, noise, and power analysis. In *DAC '08: Proceedings of the 45th Design Automation Conference*, pages 456–461, 2008.

[108] Mohamed H. Abu-Rahma and Mohab Anis. A statistical design-oriented delay variation model accounting for within-die variations. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 27(11):1983–1995, Nov 2008.

[109] Puneet Gupta and Fook-Luen Heng. Toward a systematic-variation aware timing methodology. In *DAC '04: Proceedings of the 41st Design Automation Conference*, pages 321–326, 2004.

[110] Kunhyuk Kang, Bipul C. Paul, and Kaushik Roy. Statistical timing analysis using levelized covariance propagation considering systematic and random variations of

process parameters. *ACM Transactions on Design Automation Electron Systems*, 11(4):848–879, 2006.

[111] Paul Friedberg, Yu Cao, Jason Cain, Ruth Wang, Jan Rabaey, and Costas Spanos. Modeling within-field gate length spatial variation for process-design co-optimization. *Proceedings of the SPIE (Society for Optical Engineering) 2005*, 5756:178–188, May 2005.

[112] Michael Orshansky, Linda Milor, Pinhong Chen, Kurt Keutzer, and Chenming Hu. Impact of systematic spatial intra-chip gate length variability on performance of high-speed digital circuits. In *ICCAD '00: Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pages 62–67, 2000.

[113] P. Gupta, A. B. Kahng, D. Sylvester, and J. Yang. A cost-driven lithographic correction methodology based on off-the-shelf sizing tools. In *DAC '03: Proceedings of the 40th Design Automation Conference*, pages 16–21, 2003.

[114] S. Director and R. Rohrer. The generalized adjoint network and network sensitivities. *IEEE Transactions on Circuit Theory*, 16(3):318–323, Aug 1969.

[115] A.R. Conn, P.K. Coulman, R.A. Haring, G.L. Morrill, and C. Visweswariah. Optimization of custom mos circuits by transistor sizing. In *Digest of Technical Papers from the IEEE/ACM International Conference on Computer-Aided Design*, pages 174–180, Nov 1996.

[116] D.J. Frank, Y. Taur, M. Ieong, and H.-S.P. Wong. Monte carlo modeling of threshold variation due to dopant fluctuations. *Digest of Technical Papers from the Symposium on VLSI Technology*, pages 169–172, 1999.

[117] Wei Zhao, Yu Cao, F. Liu, K. Agarwal, D. Acharyya, S. Nassif, and K. Nowka. Rigorous extraction of process variations for 65nm cmos design. In *ESSDERC 2007: 37th European Solid State Device Research Conference*, pages 89–92, Sep 2007.

[118] Anantha P. Chandrakasan, William J. Bowhill, and Frank Fox. *Design of High-Performance Microprocessor Circuits*. Wiley-IEEE Press, 2000.

[119] George Casella and Roger L. Berger. *Statistical Inference Methods*. Wiley-IEEE Press, 2000.

[120] Chandramouli Kashyap, Pouria Bastani, Kip Killpack, and Chirayu Amin. Silicon feedback to improve frequency of high-performance microprocessors - an overview. *ICCAD 2008: IEEE/ACM International Conference on Computer-Aided Design*, pages 778–782, Nov. 2008.

[121] P. Bastani, N. Callegari, L.-C. Wang, and M. Abadir. An improved feature ranking method for diagnosis of systematic timing uncertainty. *VLSI-DAT 2008: IEEE International Symposium on VLSI Design, Automation and Test*, pages 101–104, April 2008.

[122] D. Blaauw, V. Zolotov, and S. Sundareswaran. Slope propagation in static timing analysis. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 21(10):1180–1195, Oct 2002.

[123] Savithri Sundareswaran, Jacob A. Abraham, Sergey Gavrilov, Roman Soloviev, and Rajendran Panda. A timing methodology considering within-die clock skew variations. In *2008 IEEE International System-On-Chip Conference*, pages 351–356, 2008.

[124] Ben Gu, Kiran Gullapalli, Yun Zhang, and Savithri Sundareswaran. Faster statistical cell characterization using adjoint sensitivity analysis. *IEEE Custom Integrated Circuits Conference*, Sep 2008.

[125] D. Blaauw, V. Zolotov, S. Sundareswaran, C. Oh, and R. Panda. Slope propagation in static timing analysis. *ICCAD-2000: IEEE/ACM International Conference on Computer Aided Design*, pages 338–343, 2000.

[126] R. Panda, S. Sundareswaran, and D. Blaauw. Impact of low-impedance substrate on power supply integrity. *IEEE Design and Test of Computers*, 20(3):16–22, May-June 2003.

[127] Min Zhao, Yuhong Fu, V. Zolotov, S. Sundareswaran, and R. Panda. Optimal placement of power-supply pads and pins. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25(1):144–154, Jan. 2006.

[128] Min Zhao, R. Panda, S. Sundareswaran, Shu Yan, and Yuhong Fu. A fast on-chip decoupling capacitance budgeting algorithm using macromodeling and linear programming. *DAC'06: Proceedings of the 43rd Design Automation Conference*, pages 217–222, 2006.

[129] B. Cline, K. Chopra, D. Blaauw, A. Torres, and S. Sundareswaran. Transistor-specific delay modeling for ssta. *DATE'08: Proceedings of the Design, Automation and Test in Europe*, pages 592–597, March 2008.

[130] Sanjay Pant, D. Blaauw, V. Zolotov, S. Sundareswaran, and R. Panda. Vectorless analysis of supply noise induced delay variation. *ICCAD-2003: Proceedings of the International Conference on Computer Aided Design*, pages 184–191, Nov. 2003.

[131] P. Bastani, B.N. Lee, Li.-C. Wang, S. Sundareswaran, and M.S. Abadir. Analyzing the risk of timing modeling based on path delay tests. *ITC 2007: IEEE International Test Conference*, pages 1–10, Oct. 2007.

[132] Min Zhao, R. Panda, B. Reschke, Yuhong Fu, T. Mewett, S. Chandrasekaran, S. Sundareswaran, and Shu Yan. On-chip decoupling capacitance and p/g wire co-optimization for dynamic noise. *DAC '07: Proceedings of the 44th ACM/IEEE Design Automation Conference*, pages 162–167, June 2007.

[133] Min Zhao, Yuhong Fu, V. Zolotov, S. Sundareswaran, and R. Panda. Optimal placement of power supply pads and pins. *DAC '04: Proceedings of the 41st ACM/IEEE Design Automation Conference*, pages 165–170, 2004.

[134] S. Pant, D. Blaauw, V. Zolotov, S. Sundareswaran, and R. Panda. A stochastic approach to power grid analysis. *DAC '04: Proceedings of the 41st ACM/IEEE Design Automation Conference*, pages 171–176, 2004.

[135] R. Panda, S. Sundareswaran, and D. Blaauw. On the interaction of power distribution network with substrate. *ISLPED'01: International Symposium on Low Power Electronics and Design*, pages 388–393, 2001.

[136] R. Tayade, S. Sundareswaran, and J. Abraham. Small-delay defect detection in the presence of process variations. *ISQED '07: Proceedings of the 8th International Symposium on Quality Electronic Design*, pages 711–716, March 2007.

Vita

My name is Savithri Sundareswaran. I graduated with a B.E. (Honors) in Electrical and Electronics Engineering and a Masters in Physics degrees in 1994, from Birla Institute of Technology and Science, India. My interest to pursue a career in research was strong right from early graduate studies. I worked for a couple of years as a Scientist at Central Electronics Engineering Research Institute in India. In 1995, I joined Motorola India where I worked on research and development of advanced tools in transistor level analysis and optimization. In 1999, I relocated to Motorola in Austin, Texas where I continued to develop advanced and differentiating tools for power grid analysis, statistical timing characterization and analysis. In 2003, I decided to go back to school while continuing to work at Motorola. After the spin-off of Motorola's Semiconductor Sector to Freescale, I took a break from school work for over a year. While continuing my work at Freescale, I went back to finish my dissertation in 2009. I am now a Ph.D. in Computer Engineering from The University of Texas at Austin, USA.

Permanent address: 9805 Llano Estacado Ln., Austin Texas 78759 USA